DEVELOPMENT OF A MECHATRONICS EDUCATION DESK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

ONUR ALPER ERDENER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences.

____

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

____

Prof. Dr. Kemal İDER
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

____

Prof. Dr. Abdülkadir ERDEN
Supervisor

Examining Committee Members

Prof. Dr. Bülent E. PLATİN ____________________

Prof. Dr. Abdülkadir ERDEN ____________________

Asst. Prof. Dr. Burhanettin KOÇ ____________________

Asst. Prof. Dr. İlhan KONUKSEVEN ____________________

Dr. Buğra KOKU ____________________

ABSTRACT

DEVELOPMENT OF A MECHATRONICS EDUCATION DESK

Erdener, Onur Alper

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. Abdülkadir Erden

December 2003, 119 pages

In this thesis a mechatronics education desk is developed. The system developed is a low cost, versatile mechatronics education and teaching environment that aims to facilitate hands-on education of undergradutate level mechatronics students. The desk is formed of three main modules that address the needs of mechatronics education: The WorkDesk, Mechatronic Building Blocks and Experimental Setups. These parts are well designed and presented to form a complete and coordinated solution for mechatronics education.

The WorkDesk is a platform devoted to the mechatronics engineering trainee, which provides mechanical, electrical and software prototyping that enables studying, testing and parts integration for mechatronic projects. The components building up the WorkDesk are selected or developed to facilitate mechatronics design and prototyping.

Mechatronic building blocks necessary for mechatronics teaching are identified and selected to be a part of the system. In order to support these modules, low cost custom building blocks are also developed. These include, an autonomous mobile robot kit and a versatile interface board called ready2go. Demonstrative experiments with custom developed building blocks are also presented.

Two experimental setups are developed and presented in the scope of the thesis. The setups, Intelligent Money Selector and Heater/Cooler, address and demonstrate many aspects of mechatronic systems as well as aid introducing systems approach in mechatronics education.

As a consequence, a mechatronics education desk is developed and presented with many hands-on educational case studies. The system developed forms an extensible and flexible software and hardware architecture and platform that enables integration of additional mechatronics education modules.

Keywords: Mechatronics, mechatronics education, hands-on education, mechatronics experiments.

ÖZ

MEKATRONİK EĞİTİM DÜZENEĞİ GELİŞTİRİLMESİ

Erdener, Onur Alper

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Abdülkadir Erden

Aralık 2003, 119 sayfa

Bu tezde bir mekatronik eğitim düzeneği geliştirilmiştir. Geliştirilen bu sistem, lisans düzeyindeki mekatronik öğrencilerinin deneye dayanan eğitimini kolaylaştırmayı amaçlayan, düşük maliyetli, çok yönlü mekatronik eğitim ve öğretim ortamıdır. Düzenek, mekatronik eğitim ihtiyaçlarına işaret eden üç esas birimden oluşmuştur: Çalışma düzeneği, Mekatronik Yapı Taşları ve Deney Düzenekleri. Bu birimler, mekatronik eğitim için tam ve koordine edilmiş bir çözüm oluşturmak üzere tasarlanmış ve sunulmuştur.

Çalışma düzeneği, mekatronik mühendisliği öğrencisine adanmış ve mekatronik projeler üzerinde çalışma, test etme ve bütünleştirmeye olanak sağlayan mekanik, elektronik ve yazılım prototip imkanlarından oluşur. Çalışma düzeneği yapılanma bileşenleri mekatronik tasarım ve prototiplemeyi kolaylaştırmak için seçilmiş ve geliştirilmiştir.

Mekatronik öğretim için gerekli mekatronik yapı taşları sistemin bir parçası olacak şekilde tanımlanmış ve seçilmiştir. Bu birimleri desteklemek amacı ile düşük maliyetli özel yapı taşları da geliştirilmiştir. Bunlara, otonom hareketli bir

robot ve ready2go olarak adlandırılan çok yönlü arayüz kartı eklenmiştir. Özel geliştirilmiş yapı taşları kullanılarak gösteri amaçlı deneyler ayrıca sunulmuştur.

Bu tezin kapsamı içinde iki deney düzeneği geliştirilmiş ve sunulmuştur. Deneyler; Akıllı Para Seçicisi ve Isıtıcı/Soğutucu, mekatronik sistemlerin birçok bakış açısına işaret eder; mekatronik eğitimde sistem mühendisliği yaklaşımının sunulmasına yardımcı olur.

Sonuç olarak, bir mekatronik eğitim düzeneği birçok deneye dayanan çalışma durumu ile geliştirilmiş ve sunulmuştur. Sistem, ek mekatronik eğitim birimlerinin bütünleşmesine olanak sağlayan genişletilebilir ve esnek yazılım ve donanım mimarisi ve düzeneğinden oluşmuştur.

Anahtar Kelimeler: Mekatronik, mekatronik eğitimi, deneye dayalı eğitim, mekatronik deneyler.

ACKNOWLEDGMENTS

I send my kindest regards and thanks to my supervisor Abdülkadir Erden for supporting me endlessly in achieving this study. I also express my sincere appreciation to him for guiding me into to the field of mechatronics.

Thanks go to my colleagues in Aselsan for supporting me throughout the implementation phase of this study. I am grateful to my friend Murat Karataş for his help, advises, knowledge and creativity. I thank Enis Naci Özkan, İlker Gürel, Ramazan Arık, Soner Yiğit, Hakan Akbulut and the personnel of the machine shop for helping me during manufacturing and integration of the modules.

I also appreciate the feedback and support from the members of METU Robot Society. The technical suggestions as well as continuous support from my dear friends Gökay Sezer and Can Yüksel are gratefully acknowledged.

I finally express my sincere thanks to my family for always trusting and motivating me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE

CHAPTER 1

INTRODUCTION

It is a fact that in today's modern world demands of the market and customers have grown out tremendously. There is a continuous and dynamic development in the market conditions as a direct result of generation of high technology products incorporating complex and increased number of functionalities. Requirements specified for products are towards higher quality and reliability with lowered costs and decreased time to market duration. The design cycle for these products often take place in a competitive environment, where following the trends in technology itself and responding to innovative solutions from competitors create a challenging road for the engineering development process. Within this rapidly changing medium products, processes or systems need to be designed and developed satisfying both the customers and the developers. The changing nature of products resulted in new ways of coping with these requirements. This trend is also mentioned by [1], which focuses on the importance of the need to replace conventional methods and systems with new methods to achieve enhanced products with improved functionalities.

The need to replace classical design methods forced new technologies to be developed and applied in every field of the product lifecycle. This in turn fostered the formation of fusions of different engineering disciplines. The fusion generated novel engineering fields with new product design and development criteria. One of the end results of these new fields is Mechatronics.

Mechatronics involves the synergistic integration of mechanical, electronics and software development engineering approaches. It is something more than the summation of these fields; it brings a new approach especially at the design stage of the product. Mechatronics, which incorporates interdisciplinary thinking and design, is today a widely applied engineering discipline for the development of smart product and processes [2].

With the development and emerging of new disciplines for engineering design new approaches for training product designers are required. Engineers taught within specific fields cannot cope with the new interdisciplinary requirements, thus education for new multi-disciplinary fields such as mechatronics has become indispensable. In this sense engineering curricula, hence education on mechatronics around the world are being revised and lectures are about to be based on more laboratory work and hands-on experience. This renewal is not mainly focusing on the material taught, but rather on the methods applied and the development of resources that enable application of these methods. Consequently, many colleges and universities are considering how to address the need for mechatronics education [3].

Another important issue to be addressed for education is that the new graduates need to be well prepared for their initial work. This is best stated by [4] that in order for the fresh graduate engineers to be successful, cooperative and productive, they need to have mastered and succeeded through a well prepared and supplied background based on hands-on experience. It is evident that, without taking the industries ongoing trends, requirements and expectations into consideration, any mechatronic training philosophy will be far away from meeting the real world needs.

Consequently, to fulfill these needs, effort is being spent on the development of resources (experimental set-ups, hardware & software development tools etc.) for mechatronics engineering. This way, engineers having interdisciplinary knowledge and practical expertise on a variety of disciplines are to be trained.

There exists couple of methods applied and resources dedicated for mechatronics education in the literature. The methodology applied will closely depend on the resources shared or vice versa.

Learning is best achieved with practice. This is over-emphasized by a project carried on within the Autonomous Systems Lab (ASL) at the Swiss Federal Institute of Technology of Lausanne (EPFL) in Switzerland [5]. The project, called NLT (New Learning Technologies), is aiming the development of an environment, necessary tools and a competence to the problem-based learning applied to mechatronics. The project goal is based on what the psychological investigations have shown: a fact that in general people remember only about 10% of the content they read, whereas on the contrary this percentage is gradually increased to 90%, when they try and realize [90]. This important point is depicted in Figure 1.1 [5].

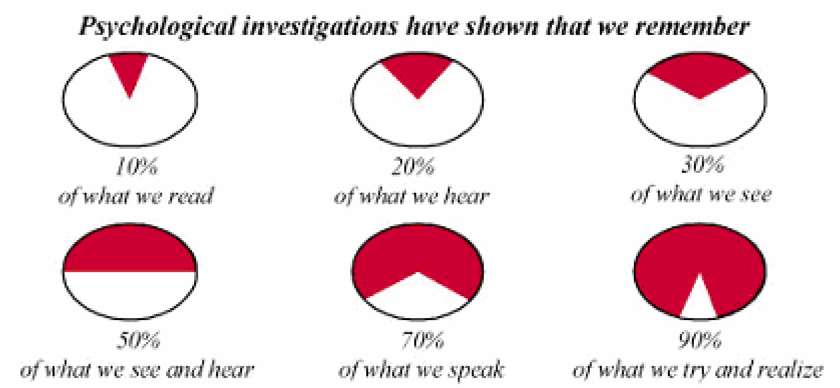


Figure 1.1 Learning efficiency [5]

It is obvious from this fact that whether the best theory given nor the variety and insight of the material taught upon any curriculum, if it is not gained by experience or practiced; it is for sure that it will be "lost" as time passes. Based on this fact, EPFL-ASL as well as other universities all over the world is aiming to adapt their curriculums for hands-on education.

The main step after sharing the idea of hands-on experience is to provide the means that will turn this into practice. In other words, the resources that will be used for hands-on education should be clearly identified. These resources, on the light of current applications, are mainly formed of the following three parts: *Facilities* containing prototyping environment, equipment (tools) and workspace, *mechatronic building blocks* composed of actuators and drives, sensors, controllers, software modules and *mechatronic experimental set-ups*.

Along with the hands-on practice idea, necessary medium and tools are to be introduced in order to make practice possible. This is achieved in a number of ways. For instance, some universities provide a laboratory completely dedicated for mechatronics ([6], [7], [8], [9]). Most of these laboratories are fully capable of providing necessary circumstances for complete production of a mechatronics system [6]. On the other hand, some departments prefer to introduce an experimental apparatus, which focuses on implementation of the readily available set-ups [10]. This way demonstrative projects serve as a method for teaching mechatronics concepts. As well as the experimental equipment, custom made hardware and software tools are also used for teaching mechatronics. These tools range from motor drivers, special sensor interfaces and module specific communication software to complex microcontroller boards. The main goal of these tools is to facilitate interfacing to the building blocks, so that complex mechatronic systems design can be accomplished in limited time and cost constraints. On the light of these, the coming sections present some of the approaches applied and tools dedicated for mechatronics education.

The Department of Mechanical Engineering at Queen's University in Canada created an approach, which involved a sequence of six laboratories culminating with an apparatus consisting of a microprocessor control of a floating ping pong ball for undergraduate laboratories in mechatronics [11].

A questionnaire based survey conducted [12] among academic and industrial Japanese engineers suggested that the educational methodology needs a "systems approach" using "systems thinking", practical training with applications to real systems, real design and manufacturing problems and hands-on experience in computer interfacing.

Iowa State University in the U.S.A presented an overview of controls taught in the mechatronics courses with issues including microprocessor input/output (I/O) constraints, digital to analog (D/A) and analog to digital (A/D) integrated circuit tradeoffs, data transfer challenges, system timing considerations, program streamlining and optimization [13]. Shetty et. al. [14] introduced a team based approach to mechatronics that attempted to make the educational experience to what the students will find in the industrial setting and emphasized on the physical understanding of sensors, actuators, data acquisition and control. The case studies within the mechatronics curriculum were implemented using general purpose I/O Board, low cost equipment, and visual simulation environment and application software. The Electrical Interface System (EIS), named Mark III, was designed in the School of Mechanical Engineering at Georgia Tech in the U.S.A. for use within early undergraduate mechatronics education [15].

Craig and Stolfi [1] proposed an integrated mechatronic approach to teaching controls to university students and professional engineers. This approach is implemented in the U.S.A at the Rensselaer Polytechnic Institute (RPI) together with the company TCG via the mechatronics education set-ups created. The Department of Mechanical Engineering at the University of Utah in the U.S.A has designed internet based labs for mechatronics education [16]. Northwestern

University of the U.S.A has created a Mechatronics Design Laboratory dedicated for mechatronics education [6].

As can be seen from the above discussions, there is no unique solution or approach for mechatronics education among universities. However, the focus is given to the formation of tools serving for mechatronics training via hands-on education.

This thesis focuses on the development of a mechatronics education desk, which will enable teaching and implementation of methods and tools necessary for mechatronics education. In the light of new technological advancements, design approaches for mechatronics will be identified. Current methods and tools applied for teaching these approaches and requirements for the mechatronics education desk will be explored. Consequently, a mechatronics education desk fulfilling these requirements will be designed and developed. The thesis proceeds as explained in the following way.

A survey on mechatronics in general is presented in Chapter 2, which includes the topics regarding to the history and definitions on mechatronics, key elements constituting mechatronic systems, architectures of mechatronic systems and advantages of mechatronic design approaches against conventional design methods. In Chapter 3, the analysis of current methods and tools used for teaching mechatronic design approaches are given. In Chapter 4, methods and tools meeting these requirements are formed and the general specifications of the mechatronics education desk together with its subsystems are exposed. In Chapter 5, experimental set-ups developed for the mechatronics education desk are explained in detail. Further and final discussions with future works to be carried on the mechatronics education desk are presented in Chapter 6. The appendices present detailed explanations on the material developed in the scope of this thesis.

CHAPTER 2

MECHATRONICS

In this chapter, results of a survey conducted on mechatronics and its definitions, properties and functions of mechatronic products, functional flow of mechatronic systems and advantages of mechatronic design approach over the conventional design methods are presented. This chapter forms the first part of the complete literature survey and prepares the background on mechatronics education suggestions and requirements.

Mechatronics was first coined from the words **Mecha**nics and Elec**tronics** in Japan in 1969 by an engineer called Tesuro Mori of Yasakawa Electric to address the merging of mechanical and electrical engineering disciplines [17]. As a consequence of the continuous and dynamic development of market conditions together with the generation of high technology products, which incorporate complex and increased number of functionalities, *mechatronics* has evolved and has been redefined for many times. A general definition states that *mechatronics* is the *synergistic* combination of mechanical engineering, electronics, control systems and computers [1]. The key element in mechatronics is the *integration* of these areas through the design process [1]. Embedding of such a process into new product development cycle introduces great advantages over classical methodologies applied, which results in generation of optimized, low cost and modular system (product) architectures. This common approach for mechatronics representing the interdisciplinary structure is shown in Figure 2.1[18].

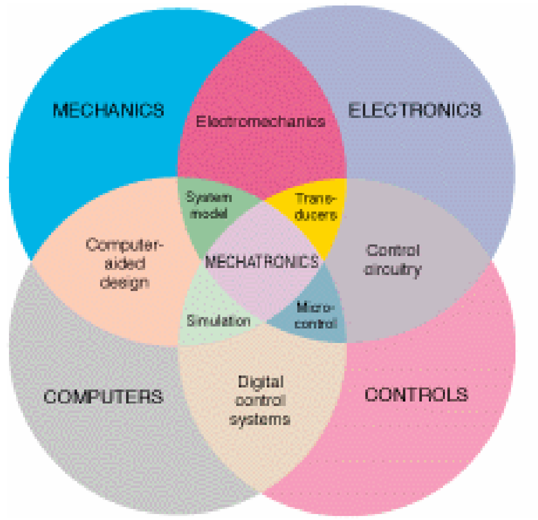


Figure 2.1 Mechatronics – A synergistic combination of disciplines [18]

Another approach into mechatronics is as presented in Figure 2.2 [14]. In this point of view information systems is presented as an integral part of the synergistic combination [14]. Mechatronics, from this perspective, includes not only the physical nature of systems, but also the surrounding information flow and/or medium. It should be noted that communication systems is also presented as an integral part of mechatronics starting from the early 90's [18]. Emphasis should be given to the fact that a physical phenomenon cannot be improved without being simulated, modeled, optimized, or appropriately controlled.

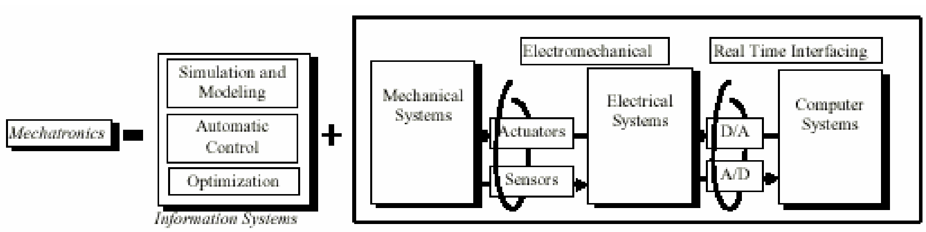


Figure 2.2 A mechatronics systems combine information systems and physical systems [14]

On the light of these, it can be summarized that mechatronics is a method applied starting from the design stage and further continued to be applied till the end product to enhance the end result (output) of a design (product/process/system) in consideration.

Among the above mentioned principle approaches various other definitions also exist for mechatronics. In order to form a firm understanding on mechatronics, concepts (keywords) underlying over 30 different definitions found in the literature are examined. The concepts derived from these definitions are grouped into similar keywords and are presented in Table 2.1. The link to the complete list of definitions for mechatronics is given in Appendix A.

Table 2.1 Mechatronics concepts

References	**Keywords**
Buur, J. [85]	Technology.
Yamaguchi, T. [74] Ashley, S. [18]	Methodology.
Shetty, D., Kolk, R.A. [30]	Combination of mechanical and electrical engineering, computer science and information technology.
Bradley, D.A. et. al. [31]	Precision mechanical engineering.
Shetty, D., Kolk, R.A. [30]	Control systems and numerical methods.
Hewit, J. R. [64] Mechatronics Research Group, UK [65]	Synergistic integration.

Table 2.1 (continued).

References	**Keywords**
Bolton, W. [29]	Bringing together.
Clemson University, USA [78]	Blending.
Virginia Polytechnic Institute, USA [87]	Unified framework.
Buur, J. [85]	Spatial integration in components, modules, products and systems.
Day, C. B. [68]	Multi-disciplinary approach. Integrated teams.
University of California, USA [83]	Multi-technological approach.
Hewit, J. R. [64]	Computer control.
Auslander, D. [71]	Application of complex decision making.
University of Washington, USA [73]	Improved product quality and performance.
Shetty, D., Kolk, R.A. [30] Loughborough University, UK [65]	Intelligent products.
Rzevski, G. [70]	Designing and manufacturing effective machines.
University of Massachusetts, USA [76]	Functionality, flexibility, and reliability.... multifunctional, interdisciplinary design and manufacturing approach.
Ashley, S. [18]	Functional and adaptable products.
Alptekin, S. [72]	Smart products.
Rizzoni, G., Keyhani, A. [19]	Sensors and instrumentation technology, drive and actuator technology, embedded real-time microprocessor systems and real-time software.

The keywords presented in the previous section show important ingredients of a mechatronic system. On the light of these concepts following inferences can be made to further analyze and deepen the meaning of mechatronics.

- Mechatronics is an engineering design discipline.
- Mechatronics involves synergistic integration from other disciplines including, but not limited to Mechanical, Electrical & Electronics, Computer, Information and Control Engineering disciplines.
- Integration through the design stage is vital as well as support (input) from other disciplines is required.
- Mechatronics (systems) philosophy, integrating resources from other disciplines is applied to output products, processes and systems (of systems).
- Mechatronic outputs, due the nature of synergy involved, exhibit improved performance, multi-functionality, and flexibility together with enhanced reliability.

Following the above analyses a further re-grouping of these concepts regarding mechatronics can be made and a more general definition for mechatronics may be given as what follows: *Mechatronics* is an engineering design philosophy applied with the synergy of disciplines to produce smart, flexible and multi-functional products, processes and systems. The re-grouping of the concepts within mechatronics is also given in Figure 2.3.

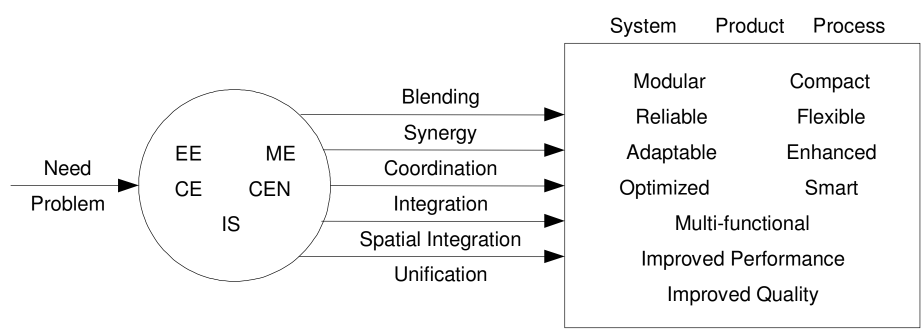


Figure 2.3 An overview on defining mechatronics

It can be stated that mechatronics is to be applied from the definition of the problem or the need. Thereafter, following a coordinated synergy among the required engineering disciplines, outputs a system (product/process) with enhanced properties over the products created via the application of classical engineering methodologies.

A mechatronic output can also be distinguished or defined in the sense of its functional flow (operation). Many mechatronic products exhibit the following general functional flow: Percept the environment, decide and affect the environment, accordingly. This is made possible in reality via the application of sensor, controller, software, and drive and actuator technology [19]. Mechatronic end products exhibit enhanced functions and improved properties over conventional designs. This is due the synergistic integration of key elements constituting mechatronics and the ways of integration applied. The key elements of mechatronics can be divided into the following specialty areas [20]:

- Physical System Modeling
- Sensors and Actuators
- Signals and Systems
- Computers and Logic Systems
- Software and Data Acquisition

The mechatronic design approach requires the application of these elements and involves hardware and software integration throughout the design lifecycle for improved performance and functions. The properties of conventional versus mechatronic design systems are tabulated below [21].

Table 2.2 Properties of conventional and mechatronic design systems [21]

Conventional Design	**Mechatronic Design**
Added components	**Integration of components (hardware)**
Bulky	Compact
Complex mechanisms	Simple mechanisms
Cable problems	Bus or wireless communication
Connected components	Autonomous units
Simple control	**Integration by information processing (software)**
Stiff construction	Elastic construction with damping by electronic feedback
Feedforward control, linear (analog) control	Programmable feedback (nonlinear) digital control
Precision through narrow tolerances	Precision through measurement and feedback control
Nonmeasurable quantities change arbitrarily	Control of nonmeasurable estimated quantities
Simple monitoring	Supervision with fault diagnoses
Fixed abilities	Learning abilities

The comparison given in Table 2.2 [21] divides the properties into two, which should not necessarily be considered as distinct parts. For instance, a mechatronic design may still involve application of simple control techniques. However, the properties grouped above may provide a guidance to have a measure of the characteristics of the design.

The most important part of mechatronic products is the flexibility devoted as a result of software integration. Mechatronic products are empowered with embedded software, which makes them extremely flexible and powerful with respect to conventional products. This is due to the very fact that, simple modulation within software can almost change the whole product function and enable configurable design, which eventually decreases the new product development life cycle. As a natural consequence of integration of multiple disciplines from within the design stage, the product is forced to be of a modular fashion, in a way forced to consist of functional modules. This modularity permits the product developers, create and improve products within short time durations that also results with a decrease in the maintainability costs of the end product.

The Traditional Design Approach – Disadvantages

The following indicates the traditional procedure applied in the development of product or systems:

- Problem/Need Formulation → Requirements are specified.
- Preliminary Design → System and subsystem interfaces are specified.
- Detailed Design → Component level specification and validation.
- Implementation (Hardware & Software) → Prototype System.
- Testing – Design Validation → Design Verification.
- Production/Manufacturing Testing → Final system for production.

This fundamental development cycle presented introduces systems to be developed in a formal way and provides going back and forth within each phase with ease, but usually lacks flexibility as processing on multiple steps back or forth is not feasible due the work power and time constraints previously determined.

This commonly applied development process also depends strictly on each phase for enabling the next phase to proceed. This is usually the case since responsible person involved within each phase use different tools. Therefore, working in parallel within phases and continuity of work is often not achieved, thus serial development is common.

Very commonly at various stages of development, tools used for design and testing are either different or do not have interfaces with each other or people using these at each stage get experienced with the use of that specific tool he or she is involved with. A very problematic situation arises with the use of different tools in the development process. Consequently, design, testing or verification involving hardware and software implementation in every phase usually requires re-work and consumes time and work power, which is spent to understand or apply the prior work done by another expert developer.

The solution here is to provide;

- the means that will enable teams at different phases of development to use similar and interconnected tools,
- necessary facilities and education that will permit individuals of teams the ability to communicate and use the tools of each other.

Remarks

In this chapter fundamental concepts regarding to mechatronics are identified. An overview on defining mechatronics is given. The key elements constituting mechatronic products are derived and properties of conventional and mechatronic products are presented. Finally, disadvantages of the traditional design approach are provided.

Remarks made in this survey will be used as inputs to the design of the mechatronics education desk. The design stage will involve the application of mechatronics design and attention will be given to take the disadvantages of traditional design into account.

CHAPTER 3

MECHATRONICS EDUCATION

In this chapter, a survey is presented on facilities, resources of mechatronics laboratories; hardware and software tools used and experiments applied in mechatronics education together with mechatronic building blocks. The section ends with the presentation of results of the survey, which lists the requirements and suggestions on mechatronics education. This chapter forms the second and last part of the literature survey.

The previous chapter has presented several definitions associated with mechatronics. Many of these suggest that mechatronics is a well formed design philosophy in the design of multi-functional products featuring improved performance with less time to market development cycles. This necessitated mechatronics to be applied in the industry for systems or advanced products development as well. In order to initiate the application of mechatronics design approaches in practice, mechatronics education has become vital. Consequently, mechatronics is now an area of involvement and specialization in most undergraduate level studies.

For achieving training on mechatronics there are many resources available, which compromise of laboratories, workbenches, experiments, hardware and software tools with corresponding mechatronics building blocks. From this point of view, the following section presents many of the undergraduate level programs, their insight and the associated resources used in mechatronics education. Facilities, experiments and tools used widely for mechatronics training in these curricula are also presented.

3.1 Facilities

Facilities are the permanent equipment and places provided to help developing, testing and verifying a system or product. What is necessary in this sense can be extracted after consideration of the general product development life-cycle. It follows that in order to develop a system one eventually needs a medium that provides the following opportunities:

- Free room to work and practice on the system,
- Room to work together within and/or with teams,
- Room for the storage of hardware and equipment,
- Space for displaying and working on experimental apparatus.

In order to make this medium effective, it is necessary to fill this place with equipment that will enable working on every phase of product development from the design stage to the end product presentation and demonstration.

Not only limited to a free room problem, as development is a continuous process it is hard to match the free times of students and their instructors. This necessitates that the medium reserved should be accessible 24 hours a day to those involved in the mechatronics curricula.

To sum up, it can be stated that a development environment with supporting equipment at all times for students is required. From this point on, this medium mentioned herein will be named as “laboratory”. The laboratory is therefore a place where the mechatronics training or practicing is to take place.

A well prepared illustration of a facility (laboratory) for mechatronics is provided at the Mechatronics Design Laboratory of Northwestern University in the U.S.A. [6]. It is an undergraduate facility serving a mechatronics and a design competition course. Microprocessor controlled smart systems design is carried in this laboratory. The laboratory consists of an electrical/software prototyping area and a mechanical prototyping area with machine tools and hand tools. The layout of the laboratory is given in Figure 3.1.

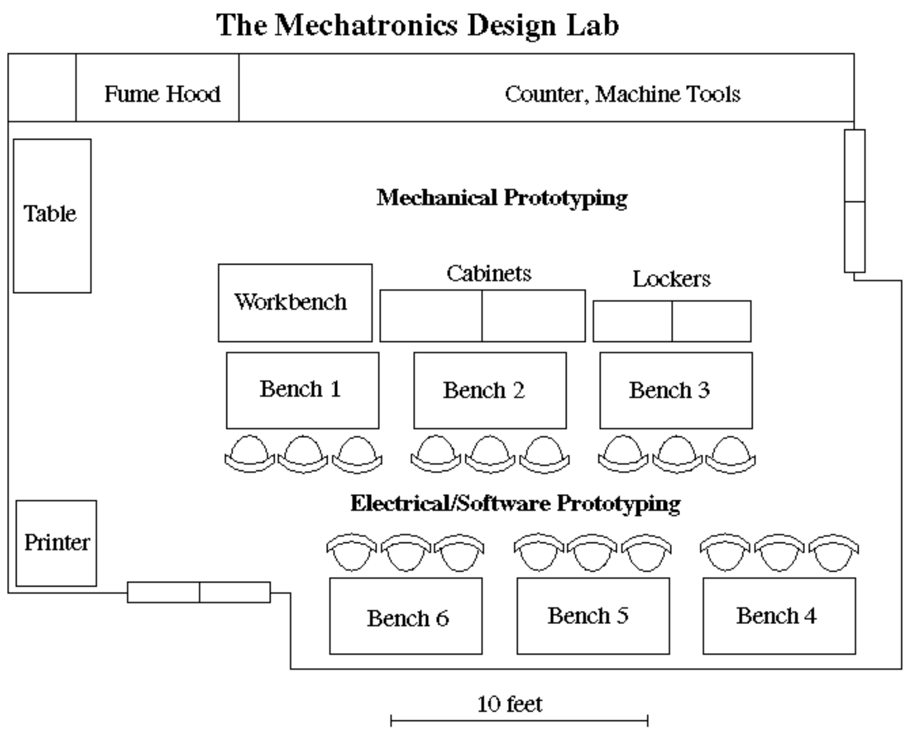


Figure 3.1 Northwestern University Mechatronics Design Laboratory layouts [6]

An important issue is to note that although a machine shop with professional technicians exists a small size prototyping area is included within this laboratory. A method applied here is to use the more sophisticated machine shop for those materials that can not be manufactured within the laboratory. The main reason for this is to provide the means to make students work 24 hours a day (especially at times when the machine shop is closed) and speed up prototyping and project development phases.

The resources within this laboratory allow a mechatronics student to work on systems design and build prototypes without any outside help. As most of the items for building a mechatronic system and prototyping are made available the overall development phase as well as the initial start up time for implementation and prototyping are gradually decreased. The student teams consisting of three individuals all share an individual workbench for electrical and software prototyping. A view of the workbench is given in Figure 3.2.

Figure 3.2 Electrical/software prototyping workbench [6]

Several similar laboratories exist as used in the "2.737 Mechatronics" course at the Department of Mechanical Engineering of Massachusetts Institute of

Technology (MIT) in the U.S.A. [7]. This course houses a new mechatronics laboratory with a wide range of facilities and serves as the main bases for laboratory work carried within the course schedule. There are 12 workstations with an additional station for the teaching assistant. The following is a view from the laboratory.

Figure 3.3 The mechatronics laboratory at MIT & an individual workstation [7]

This course teaches the design of mechatronic systems and laboratories form the core of the course. This laboratory, with its sophisticated equipment, enables complex systems design involving real time implementations. However, it lacks a mechanical prototyping area for parts manufacturing or prototyping.

The Undergraduate Teaching Laboratory at the University of Waterloo in Canada [8], consists of 8 Pentium based computers and various lab equipment available for systems design students. Throughout the undergraduate program, the Undergraduate Teaching Laboratory provides an environment for the students to perform various laboratory exercises as part of their core courses. Similarly, the "Mech Lab", located at the University of Utah in the U.S.A. [9], is a student laboratory full of computers, electronic components and experimental setups. Students come to the laboratory to work on experiments and their robot projects. As can be seen, although these laboratories provide a wide range of electronics equipment, again they lack a mechanical prototyping area.

Figure 3.4 Undergraduate Teaching Laboratory at the University of Waterloo [8]

Figure 3.5 The "Mech Lab" at the University of Utah [9]

The following station with the specifications given is available at the mechatronics curriculum of RPI [22] and the TCG [23].

- PC with MATLAB, Electronics Workbench and Working Model software.
- Function generator, digital oscilloscope, multimeter.
- Powered protoboard.
- Microcontroller, assorted analog/digital sensors, actuators and components.

Figure 3.6 A mechatronics education station from RPI and TCG [22]

As presented in the Introduction section, EPFL [5] is trying to introduce a complete solution to the mechatronics education problem. The proposed "Pedagogical Scenario" is composed of 5 steps as stated below:

- <u>Step 1:</u> Individual Web-based learning, where an internet based medium is formed to get students of different levels to an equalized level, and provide an environment where students gather and add information.
- <u>Step 2:</u> Two-day interdisciplinary workshop, where interdisciplinary teams are guided to build a simple system, aiming to form collaboration among student teams and get them realize a "system".
- <u>Step 3:</u> Individual experience with real hardware, where students practice using real hardware with the experimental kits involving basic building blocks provided.
- <u>Step 4:</u> Control experiments on a 3D robot simulator, where prior simulations on the robot system are carried on.
- <u>Step 5:</u> Robot design and final competition, where the learned material is applied on an actual robot system within a competition.

An inference from the EPFL [5] case is to use the Web as a facility or medium, which provides easy and fast access to information at all times.

An overall look at the laboratories and generalization of their specifications show that the following opportunities for students are mainly intended to be provided:

- Working space on a project,
- Equipment for systems development,
- Environment for mechanical, electrical and software prototyping,
- Space for experimentation and working on experimental equipment,
- Support for course laboratories.

As can be seen from these, the facilities directly affect the product development phases and great emphasis should be given on this issue.

3.2 Experimental Apparatus

Having an idea on existence of a system for a student, later on may lead to a gradual decrease of initial search time on a project. Also, an efficient way of teaching and experiencing new concepts is experimentation or in other words hands-on practicing. Considering all of these facts, we can derive that these can be accomplished on readily available experimental set-ups. These apparatus, mostly designed for specific tasks, can provide the means for teaching the details of many specific tasks. Following section will provide an insight on experimental apparatus being used by the mechatronics community in experimentation based mechatronics education.

It is possible to mention that an experimental apparatus covers a large content of mechatronics curricula ranging from thermal and fluid system applications to electrical systems control ([7], [8], [9], [22]). For example, the following experimental apparatus is available at the mechatronics curriculum of RPI [22] and the TCG [23], which provide education on mechatronics for universities and people coming from the industry.

Table 3.1 Experimental apparatus at RPI and TCG [23]

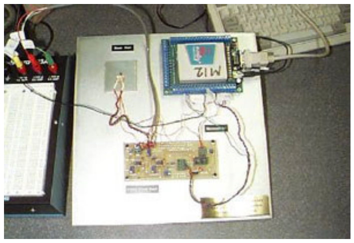
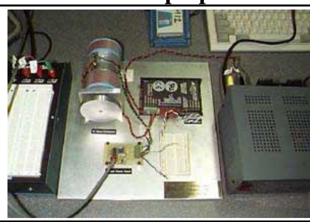
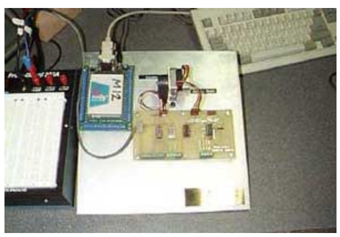
Thermal System Closed-Loop Temperature Control	**Features**
	- Aluminum plate - Thin-film resistive heater - Ceramic insulation - Conduction and convection heat transfer - AD590 temperature sensor - Microcontroller - On-off closed-loop control with relay - Support analog electronics
DC Motor Closed-Loop Speed Control	**Features**
	- Permanent-magnet brushed DC motor - Integral analog tachometer - Aluminum disk load inertia - PWM power amplifier - 24-volt, 4-amp power supply - Analog control design and implementation: lead, lag, lead-lag
Stepper Motor Open-Loop and Closed-Loop Control	**Features**
	- Stepper motor - Optical encoder - Microcontroller - Electronics to interface the microcontroller to the motor and the encoder - Full-step and half-step operation - Control via a Quad-Darlington IC - Control via a step-motor-driver IC - Programming in Basic or C
Pneumatic System Closed-Loop Position Control	**Features**
	- ¾" bore, double-acting, non-rotating air cylinder - Linear potentiometer to measure mass position - 30 psig air supply - Two flow-control valves - Two 1/8"ported, 3-way, spring-return, two-position - Solenoid valves - Darlington switches to energize solenoids - Microcontroller - On-off, modified on-off, PWM closed-loop control

Table 3.1 (continued).

Micro control Test bed	**Features**
	- Two embedded microcontrollers from MicroChip Inc. configured for: 3 channel 8-bit analog / digital (A/D), acquisition , 10-bit pulse width modulated (PWM) drive, serial communication to PC, general purpose digital I/O - High power H-bridge for output stage of pulse width modulated (PWM) driver (for d.c. motors) - Hex keypad for data entry - Liquid crystal display (LCD) for data display - Analog electronics (op amps) for measuring tachometer and input reference signal

In addition to the above figured experiments below listed test systems are also available;

- 2 mass, 3 spring dynamic system
- MR fluid rotary damper system
- Magnetic levitation system
- Control development test bed
- Spring-pendulum dynamic system

Additionally, students have developed the following experiments within the Mechatronics Systems Design [22] course:

- Hydraulically-balanced beam system
- Inverted pendulum systems
- Ball-on-beam balancing system
- Ball-on-plate balancing system

Another example of practical application of mechatronics can be found at MIT [7], where training is centered on the laboratory experience. The new laboratory facilities feature digital signal processing based – dSpace [27] processors used in real-time simulations, which are programmed through Matlab/Simulink [54]. Each student builds circuits on a breadboard kit as shown in Figure 3.7.

Figure 3.7 Breadboard kit for prototyping and dSpace interfacing used at MIT [7]

dSpace and model based design approach through Simulink is applied to eliminate low-level programming and ensure that the students concentrate on the higher level design tasks. Each student is assigned one of these project kits for prototype work during the semester. The topics covered within the laboratories are as following:

- Pre-lab 0: Review
- Lab 1: Servo Motor Control
- Lab 2: Sampling/Aliasing; Signal Processing
- Lab 3: Digital Logic; Quadrature Encoders
- Lab 4: Analog Power Amplifier Design
- Lab 5: Brushless Motor Commutation and Control

Research at the Design of Mechatronic System Research Group of the Katholieke Universiteit Leuven (KUL) [24] in Belgium does not end up at the computer simulation stage. In order to promote their practical usage, they are tested with following facilities:

- 4 dSpace DSP real time control systems for PC (ds1102 and ds1103) with software development environment.
- Heim DiSC data acquisition system.
- Two experimental test set-ups with linear motors, one equipped with air bearings, and the other with rolling elements bearing.
- Linear motor based H-drive (Philips flexcell II).
- Linear motor based X-Y table for high speed milling demonstrator (gantry design, developed in Brite-Euram BE96-3424 Motion project).
- Rotary motor ball-screw driven table.
- Flexible robot link with DC direct drive torque motor.
- Experimental cam setup (an experimental model of the main drive system of a weaving machine).
- Pneumatic servo positioning system.

There are also companies working in the field of mechatronics education, where they provide considerable professional experimental equipment. One of these firms is Feedback Instruments Limited [25]. The firm provides a Mechatronics Project Kit as shown in Figure 3.8.

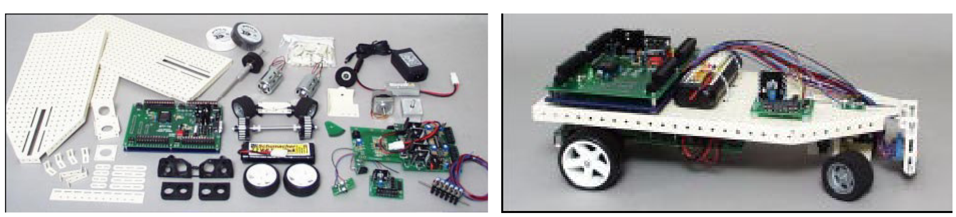


Figure 3.8 Mechatronics Project Kit of Feedback Instruments Limited [25]

The Mechatronics Project Kit compromised of different mechatronics modules provides the means for students to design an autonomous vehicle. The modules provided within this project kit are listed below:

- Microprocessor control board with PIC16F877.
- DC motor drive circuitry: single or dual motor arrangement.
- Stepper motor drive modules: Ackermann or single wheel steering arrangements.
- Different sized chassis and pivot wheel.
- Ni-Cd battery and battery charger.

Another, company in this field is Mechatronic Systems, Incorporated in U.S.A. [26]. They provide a Mechatronics Control Kit (Model M-1) as shown in Figure 3.9.

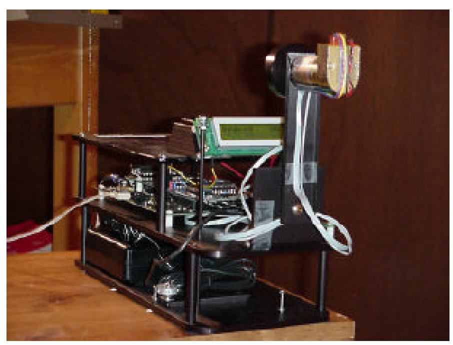


Figure 3.9 Mechatronics Control Kit of Mechatronic Systems, Inc. [26]

This kit features a wide range of applications for mechatronics control research and education. The following experiments can be handled with this kit:

- System identification,
- Friction compensation,
- Position and speed control,

- Balancing of an inertia wheel pendulum,
- Swing up and balancing of pendulum.

The control algorithms to be applied include linear control, nonlinear control, optimal control, learning control, robust and adaptive control, fuzzy logic control, intelligent control, hybrid and switching control and gain scheduling. The parts delivered with the kit are given below:

- DC motor with 1000 Cnt/Rev optical encoder.
- C6711 DSK Board and Code Composer Studio software.
- Power supply.
- LCD screen.
- C6XDSK_DIGIO daughter board (Quadrature Encoder Input, PWM Output, Digital I/O, MAX232 interface and expansion header).
- PWM amplifier board.

A very important feature of the kit is its compatibility with Real Time Workshop from Mathworks. This way real time Hardware In-the-Loop experimentation is made available from within Matlab/Simulink models.

dSpace [27], another company supplying educational and industrial tools, provides Advanced Control Education Kit (ACE Kit) for universities. The ACE Kit is a real-time development system with simulation hardware and software tools for rapid control prototyping. The hardware tools include a selection from two Real-Time Controller Boards, DS1104 R&D and DS1103 PPC, respectively. The software package includes Implementation Software that provides the Real-Time Interface with Matlab/Simulink and Real Time Workshop and the Experiment Software that contains ControlDesk and MLIB/MTRACE. ControlDesk enables to control, monitor and automate Simulink and real-time experiments, whereas the MLIB/MTRACE interface enables access to the Matlab platform and provides automation of experiment control from Matlab.

A well known company supplying mechatronics education tools is Quanser [28]. The company provides experiments to implement and evaluate various feedback strategies, such as PID, LQC, H infinity, fuzzy, neural nets, adaptive or nonlinear controllers. The experiments are made available via the linear, rotary, specialty and custom made modular components that can be configured for different experiment configurations. These configurable modules permit additional functionality and extensibility in the experiment domain.

We note that, although a great know-how can be gained from the application of experiments, without experiencing an actual mechatronic systems design process, a complete understanding of a "mechatronics system design" cannot be achieved. Therefore, we conclude that some demonstrative or experimental equipment should be included within a mechatronics laboratory, but focus must be given on the development of systems in practice.

3.3 Mechatronic Building Blocks

The Mechatronic Building Blocks (MBB) constitutes the main bricks for building up a mechatronic system. They are generally grouped into three main categories actuators, sensors and controllers (microcontrollers), respectively as mentioned in the functional decomposition of a mechatronic system. Their selection and usage is very important for a mechatronic system to be capable of being successful in completing its goal. This requires attention to be given on their selection and interfacing.

Within the scope of this thesis a survey has been conducted on the material used as mechatronic building blocks within mechatronics curricula and information regarding to this field are inspected in mechatronics related books. The coming part presents the results of this survey.

Table 3.2 List of mechatronic building blocks used for mechatronics education

Mechatronic Building Blocks	**References**
Actuators	
Hydraulic Actuators	[29,30,32,33,34,35 and 38]
Pneumatic Actuators	[29,33,34,35 and 38]
DC Motors	[30,31,32,33,35,37,39,43, and 44]
Brushless Motors	[31,32,37 and 39]
Actuators	
Step Motors	[29,31,32,33,34,35 and 39]
RC Servomotors	[43 and 44]
Solenoids	[29,31,32,33 and 40]
Piezoelectric Actuators	[30,31 and 41]
MEMS	[42]
AC Machines	[3,4 and 39]
Shape Memory Alloys (SMA)	[45]
Sensors	
Spatial Variables Measurement [11] Sensors	[31,32,35,36,37,46,47 and 48]
Time and Frequency Measurement [11] Sensors	[46]
Mechanical Variables Measurement (Solid) [11] Sensors	[30,31,32,33,34,36,43 and 46]
Mechanical Variables Measurement (Fluid) [11] Sensors	[32,33,36,46 and 50]
Mechanical Variables Measurement (Thermal) [11] Sensors	[30,31,32,33,36,46 and 49]
Electromagnetic Variables Measurement [11]	[46]

Table 3.2 (continued).

Mechatronic Building Blocks	References
Processors/Controllers	
Microprocessors	[29,31,32,33 and 36]
Programmable Logic Controllers (PLC)	[29,32,33 and 51]
Embedded Computers	[33 and 51]

The Embedded Computers mentioned in the table include the hardware platforms of microcontroller-based systems, Field Programmable Devices such as Field Programmable Gate Arrays, Digital Signal Processing systems and real-time platforms.

The information provided in the table show that there is no common agreement on selecting or using the mechatronic building blocks. This is an expected result as mechatronics covers a wide range of applications. Additionally, such a large content cannot be provided with practice in limited time and budget constraints of current mechatronics curricula. From an educational point of view, we can state that in order to cover the theory and practice of the listed items we need a well defined approach. A solution to this can be to implement mechatronics education of building blocks through distributing items (MBB's) within different courses and laboratories at different levels of the complete educational curriculum. This necessitates coordination among lectures and laboratories. An example for such an application can be to determine the content of concepts and tools introduced in lectures other than mechatronics and provide the remaining parts within the mechatronics curriculum. This way, the number of building blocks presented and their applications can be further increased.

In addition to the previously mentioned basic mechatronics building block types, many researchers or companies develop tools that simplify the integration of these blocks, which also enable easy and fast mechatronics application building. These tools include, but are not limited to custom developed microcontroller boards [11, 52, and 53] that provide easy interfacing with various external peripherals such as motors and sensors, embedded systems or mobile robot kits. The following section will provide detailed information regarding to the autonomous mobile robot kits.

Mobile Robots

Mobile Robots has played a key role in introducing students with real world experience. Working with mobile robots challenged many educators with its many aspects as many as system integration, multidisciplinary working and cooperation. Other reasons for using mobile robots are [62];

- Students are often fascinated by robots and respond enthusiastically to the design task.
- Building robots requires the student to use concepts learned in a number of courses in the curriculum.

There are a number of educational Mobile Robots available in the literature either developed by researchers at educational centers or made available as robot kits by companies working in the field of mechatronics. Following part introduces the most commonly used mobile robot kits.

The smartROB [5] is a custom made cheap and flexible mobile robot kit developed for the “Smart Mechatronic Product Design” course and is being used since 1994 in student projects. SmartRob kit includes several analog and digital I/Os, a Frame Grabber board for camera output processing and a PowerPC CPU board able to run real time software.

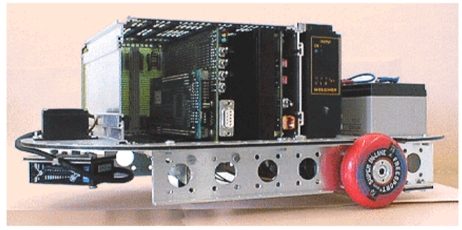


Figure 3.10 smartROB [5]

Another mobile robot kit, named Mechatronics Project Kit [25] of Feedback Systems is aimed to provide the means for students to develop their skills in mechanical & electrical design, PIC programming and project management. The kit includes PIC based microprocessor board, several optical and magnetic sensors, power electronics for DC and step motor driving and two selectable modular vehicle chassis. The Mechatronics class at the University of Utah uses the custom made mobile robot Mobius [55] for teaching fundamentals of mechatronics. The robot is equipped with Handy Board microcontroller board and 3 simple sensors to teach and exhibit simple behaviors such as object detection/avoidance, light detection and navigation.

Figure 3.11 Mechatronics Project Kit [25] and Mobius [55]

Another well known autonomous robot designed for educators, researchers, and hobbyists is the Rug Warrior Pro Kit [44], which is especially designed to accompany the famous fundamental book "Mobile Robots: Inspiration to Implementation". The kit supplies all the sensors, electronics, and mechanical hardware necessary to construct a custom designed, fully autonomous mobile robot that enables various behavior implementations. Parallax, famous with its Basic Stamp microcontrollers also supplies the Boe-Bot [56] mobile robot kit as an add-on to the Stamps-in-Class robotics curriculum. Boe-Bot kit is mainly centered on the Parallax Board of Education, an educational Basic Stamp II microcontroller board and is equipped with modified RC servo motors, IR LEDs and receivers, photoresistors and microswitches, which are used to teach light sensitive navigation, object detection, distance determination and tactile navigation concepts. The remaining survey results for mobile robots are given in Table 3.3.

Figure 3.12 Rug Warrior Kit [44] and Boe-Bot [56]

Table 3.3 Mobile Robot Kits

Robot	Actuators/Drive Type	Sensors	Processors	Software	Applications
smartRob [5]	DC & Step Motors. Differential Drive.	A variety ranging from encoders to cameras.	VME Motorola PowerPC 604 & Power Board, PMC Frame Grabber. VME PIO and ADDA can be included.	Xoberon Real-Time Development Environment.	Smart Mechatronic Product Design course, student projects and competitions.
Mechatronics Project Kit [25]	Geared DC & Step Motors. Differential Drive w/ DC motors, Ackermann, steering w/ the step motors.	Optical and Magnetic (Hall Effect) Sensors.	PIC 16F877	MPLAB.	Predetermined circuitry following.
Mobius [55]	DC motors (modified RC servos). Differential Drive.	Polaroid sonar, IR photo detectors, Vector 2x digital compass.	Handy Board	Interactive C.	Light detection, object detection/avoidance, navigation.
Rug Warrior Kit [44]	Geared DC motors. Differential Drive.	Microswitches, IR detectors, photoresistors, encoders, microphone	Motorola MC68HC11 microcontroller.	Interactive C.	Obstacle avoidance, dead-reckoning, light seeking, navigation, sound creation etc.
Boe-Bot [56]	Geared DC motors (modified RC servos). Differential Drive.	IR LEDs and receivers, photoresistors and microswitches.	Basic Stamp II (Board of Education)	Pbasic.	Light sensitive navigation, object detection, distance determination, tactile navigation.
Libby [57]	Geared DC motors (modified RC servos). Differential Drive.	IR LEDs and receivers, microswitches.	Custom Basic Stamp II Board	Pbasic.	Basic mobile robot kit allowing IR communication and simple navigation.
Mech-Bots [58]	Geared DC motors. Differential Drive.	IR LEDs and receivers, photoresistors, RF transmit/receive system, Vector 2x digital compass and temperature sensors.	Handy Board and custom PCB board for the sensors.	Interactive C.	Navigation, light following, communication, temperature measurement.

Table 3.3 (continued).

Robot	Actuators/Drive Type	Sensors	Processors	Software	Applications
ATRV [59]	Geared DC Motors. 4 wheel differential drive.	Sonar, stereo vision system, PTZ vision system, laser scanner, electronic compass, absolute orientation sensor, INS, GPS/DGPS, ASCII to Speech RS-232 Box.	Pentium Based System.	Dependent on the HW.	Rugged and reliable mobile robot for most demanding outdoor all-terrain projects.
Magellan Pro [59]	2 DC servomotors. Differential Drive.	Sonar, IR, microswitches, vision system, wireless communication, laser scanner, electronic compass, text to speech system.	Onboard Pentium III.	Mobility Robot Integration Software	Advanced Mobile Robot Platform.
B21r [59]	4 High Torque DC servomotors. Synchronous Drive.	Same with Magellan Pro.	Up to 3 Onboard Pentium III Boards.	Same with Magellan Pro.	Advanced Mobile Robot Platform.
Palm Pilot Robot Kit [60]	3 Geared DC motors (modified RC servos). Omni-Drive.	3 Range Sensors (Sharp GP2D12).	Palm Pilot.	Code Warrior, any Palm Development Environment.	General purpose robot kit enabling obstacle avoidance minimum.
Carpet Rover Robot Kits [61]	Geared DC motors (modified RC servos). Differential Drive.	Microswitches, IR proximity detector kit, optical line tracking kit.	Basic Stamp 2.	Pbasic.	Line tracking, light following, light avoiding, robot art, maze or wall following, obstacle avoidance.

3.4 Mechatronics Education Requirements/Suggestions

On the light of the information gathered through the literature survey conducted many requirements and suggestions for mechatronics education are derived. These suggestions are brought together and presented in the coming part. As the main goal of the theses a mechatronics education desk will be developed to meet these requirements. As a starting point for requirements analysis we can state the following remarks as general specifications or main points to take into account in mechatronics education:

- Focus should be given on integration of multiple disciplines and mechatronics systems thinking.
- Mechatronics requires interaction with the environment. Therefore, the use of real world examples and applications in education are essential.
- The tools (i.e., software) selected for mechatronics education should be well integrated to each other to enable data exchange. Another requirement is to determine tools that will permit working in parallel all through the design cycle.

Requirements for Facilities

This section summarizes the requirements for mechatronics education facilities.

Table 3.4 Requirements for mechatronics education facilities

Requirements Suggestions	**Remarks**
Laboratory	There is an indispensable need for free room to work on a project.
24 hours laboratory access	The laboratory should be accessible 24 hours a day to the students.
Web based facilities	In order to enhance access to the laboratory resources, experiments, and reading material as well as to enable collaboration within teams there is a need to establish a web based resource environment.
Prototyping areas	The laboratory should accommodate electrical, mechanical and software prototyping areas.
Components	The laboratory should incorporate many components (electrical and mechanical) that will enable prototyping of mechatronic designs.
Equipment & Tools	The laboratory should have equipment and tools that enable electrical and mechanical prototyping. This equipment should include basic hand tools and even machine tools such as drills and mills for simple parts manufacturing. The tools involved should include computers, printers, oscilloscopes, function generators, software tools etc.
Workbench	The laboratory should have workbenches dedicated to individual students or student teams that will permit prototyping and studying.
Support for course laboratories	The laboratory and its equipment should be coordinated to provide support for course laboratories and should not be limited to mechatronics design.
Experimentation & experimental set-ups	The laboratory establishment should enable hands-on practicing and material that provide experiments.

Requirements for Experimental Set-ups

This section summarizes the requirements for mechatronics education experimental set-ups.

Table 3.5 Requirements for mechatronics education experimental set-ups

Requirements Suggestions	**Remarks**
Experimental Set-ups	There should be sufficient number of experimental setups that would cover a wide range of mechatronics topics. These fields range from, thermal and fluid systems to electromechanical systems. The experiments should be selected to demonstrate fundamental concepts as well as new leading edge technologies.
Involvement in experiments	The experimental setups should provide the students to involve deeply during experimentation.
Hands-on practicing	The experimental setups should allow hands-on education of students.
Mechatronic Building Blocks	Focus should be given to introduce students with different mechatronic building blocks in the experimental setups.
Real hardware-Industrial perspective	The experimental setups should be preferred to be formed of real hardware used in the industry to let students familiarize with real world applications.

Requirements for Mechatronic Building Blocks

This section summarizes the requirements for mechatronic building blocks used in mechatronics education.

Table 3.6 Requirements for mechatronics education mechatronic building blocks

Requirements Suggestions	**Remarks**
Mechatronic Building Blocks	The students should be provided with a large pool of mechatronic building blocks containing assorted microcontrollers, sensors and actuators. The building blocks should be carefully selected to cover a wide range of possibilities for hands-on education.
Support equipment	In order to simplify mechatronics design and development and reduce time spent on the project, equipment that enables easy interfacing to the building blocks should be provided. This equipment may be formed from readily available components or can be custom developed.
Robots	Robots (i.e., mobile robots) attract students and provide a large content in the mechatronics applications. Hands-on practicing via robots is an enabling method for mechatronics training.

General requirements depicted above must be well bonded in order to perform a complete and dedicated mechatronics education. The bonding can be via collaboration of students within applications, a design project and competition of designs. The bonding should also have a systems thinking point of view.

The remaining part of the thesis will denote the development of a mechatronics education desk on the light of the material presented.

CHAPTER 4

MECHATRONICS EDUCATION DESK

In this chapter, specifications for the Mechatronics Education Desk are given. The content of MED is provided and details of parts developed are explained. As to form a basis for the educational desk, requirements gathered from the literature survey are also presented where applicable.

4.1 Scope of MED

The development process of Mechatronics Education Desk will address the following requirements mentioned previously in Chapter 3.

- Determination of the requirements for the mechatronics design laboratory.
- Determination of mechanical, electrical and software prototyping environment.
- Determination of mechanical and electrical components to be used in the laboratory.
- Determination of the individual workbench.
- Development of sample experimental setups addressing mechatronics system interfacing (integration) and real-time control. Attention will be given to provide applications from the real world practice.
- Selection of Mechatronic Building Blocks to be made available in the laboratory.
- Development of an interface board that will facilitate the use and interfacing of mechatronic building blocks.

- Development of an autonomous mobile robot kit.

4.2 Definition of MED

MED is a low cost, versatile mechatronics education environment developed for enabling hands-on practicing of undergraduate mechatronics students.

MED is formed by a WorkDesk containing mechatronic key elements such as Mechatronic Building Blocks composed of assorted actuators and drives, sensors, controllers and software tools, commonly used mechanical and electronic elements necessary for building up a mechatronic design and custom developed experimental setups.

MED is considered to be a part of a Mechatronics Design Laboratory (MDL). MDL and MED are two are supporting environments for mechatronics education. Components that are not contained within MED are considered to be provided from the MDL as centralized common supplies. A general view of MED is provided in Figure 4.1.

Figure 4.1 A view from Mechatronics Education Desk

4.3 Aim

MED is aimed to be used as a WorkDesk and as a hands-on education medium (tool) dedicated to the mechatronics trainees of undergraduate level studies. Mechatronic design approaches and concepts are aimed to be practically and easily taught with the aid of MED. Additionally, the following two main goals are addressed:

1. Providing a means to learn and experience mechatronic building blocks.
2. Providing a means for developing mechatronics design and prototyping of mechatronic projects.

Other sub goals to be fulfilled in this scope include;

I. Providing the students with various tools starting from the early design stage going through modeling, prototyping and testing phases towards the mechatronic end product.

II. Providing a medium for converting theoretical background gained from lectures into practical means.

III. Providing the students with enough hands-on experience so that they can be productive just after graduation.

Another key role considered for MED is its benefit for providing students to concentrate on their actual design task. Students are no longer required to discover concepts or materials that are common, easy to reach and have obsolete know-how. Many mechatronic components and building blocks are served with easy interfaces in the mechatronics education desk. This way, students are rather fostered to focus on mechatronic systems, systems thinking, system integration and interfacing. The main contribution is in accomplishing a medium that combine various mechatronic hardware and software resources towards achieving educational and research goals.

4.4 The MED System Architecture

MED is composed of three main parts, the WorkDesk, Mechatronic Building Blocks and Experimental Setup, respectively. Another category, not included within these mentioned parts, involves a facility requirement list for the Mechatronics Design Laboratory, which involves additional supporting equipment for MED. Information on this list is provided in Appendix A.

The main parts of MED are merged to form the system architecture of the Mechatronics Education Desk given in Figure 4.2. Shaded blocks reflect the parts designed and developed in the scope of the thesis, whereas other blocks correspond to readily available items searched and selected from, within the market, common of-the shelf (COTS) products.

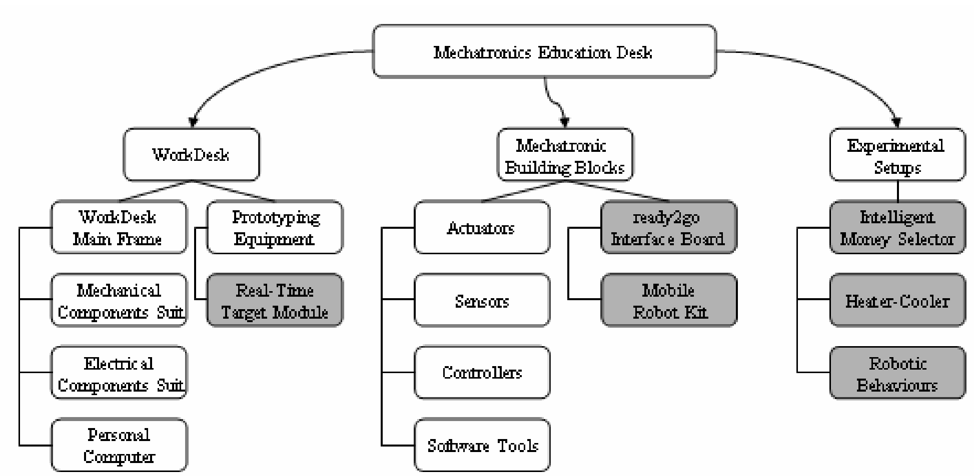


Figure 4.2 Mechatronics Education Desk block diagram

The coming sections contain the component level specification of the main three constitutes, The WorkDesk, Mechatronics Building Blocks and Experimental Setups that build up MED.

4.5 The WorkDesk

The WorkDesk is the platform devoted to the mechatronics engineering trainee. It is the main working area for a hands-on education. The WorkDesk enables mechanical, electrical and software prototyping as well as creates a suitable medium for studying, testing and parts integration. The important role of the WorkDesk is to provide the trainee with easy access to the prototyping equipment. Figure 4.3 given below shows the WorkDesk and its components.

Figure 4.3 The WorkDesk

The subsystems of WorkDesk are listed below:

- WorkDesk Main Frame.
- Mechanical and Electrical Component Suits.
- Prototyping Equipment.
- Personal Computer.

- Real-Time Target Module.

Detailed components lists regarding to the mechanical and electrical components, software tools together with the prototyping equipment and the personal computer are given in the accompanying Mechatronics Education Desk Compact Disc (MED CD) of the thesis text. These materials are determined from the survey conducted, experiences gained both from the industry and the lectures and hands-on experience within the university. Another important item of the WorkDesk, which will form the real-time hardware platform, is the Real-Time Target Module. Details corresponding to this module are explained in the next section.

4.5.1 Real-Time Target Module (RT-TM)

Mechatronic products need to interact with the environment. The environment, in another way the real world is sensed and eventually an action is taken inside the medium. As mechatronic systems have complex functions and interactions with the environment they generally require real-time response. This is especially indispensable when complex systems with closed loop control are considered. Therefore, real-time control and respective applications place a great role in the field of mechatronics. It is also another important step for providing the students to what they will observe in the industry.

Real-time control refers to the act of deterministically performing a control task without having an interruption from within other tasks. This refers to a response to the events at predetermined time steps, neither slower nor faster, but at the time defined. Real-time control does not necessarily refer to high speed control, but rather reliable and consistent response to events.

Recent advances in technology required new methods to be applied in mechatronic systems that need real-time control. These include the introduction of graphical based design tools that enable the complete design and development cycle to be fulfilled via the model based design approach. Such tools provide both non-real time simulations as well as automatic code generation for real-time target platforms, where hardware in-the loop processing is made available. One of the most common of these tools is the products of Mathworks. Models created in Simulink or Stateflow can be converted into customizable C code, where after a build process the code can be run on different real-time target platforms, such as computers running VxWorks [88] real-time operating system, digital signal processing (DSP) processors, or desktop computers with xPC Target kernel. Another similar tool is introduced by Telelogic, which is named as Tau 2 Generation [89]. This model based design software tool generates automatic code for the VxWorks target.

In the scope of undergraduate studies and considering the use of Mechatronics Education Desk in real-time application development, a Real-Time Target Module (RT-TM) is integrated to the system. The Real-Time Target Module is a custom integrated platform dedicated for real-time control prototyping. It is aimed for desktop use in mechatronic applications requiring real-time processing. The target module is developed to be compatible with the Mathworks xPC Target real-time applications. It consists of the following hardware and software subsystems given in Table 4.1.

Table 4.1 RT-TM components

Real – Time Target Module Components	
Hardware	**Remarks**
Motherboard	Onboard VGA and sound with a minimum of 2 PCI slots.
CPU	Pentium I and over.
RAM	> 16 MB
Hard Disk Drive	> 32MB
3.5” FDD	
Ethernet Card	Compatible with xPC Target.
Power Supply Unit	> 300W
Computer Case	Compatible with full PCI boards.

Table 4.1 (continued).

Real – Time Target Module Components	
Hardware	**Remarks**
Data Acquisition Board (DAQ)	Decision Systems International 12bit resolution.
Switching Unit	Provides single display, keyboard and mouse use with two computers.
Software	**Remarks**
Operating System	MS-DOS is sufficient.
Mechatronics Education Desk Simulink Library	The library contains xPC Target drivers for Parallel Port and the Data Acquisition Board.

The RT-TM does not have a display attached, but rather a switching unit is included in the system. This unit allows switching the keyboard, mouse and the display between the personal computer and the RT-TM. This way, it is made possible to increase space dedicated for prototyping and decrease the overall cost of the system. The hardware configuration for executing real-time code on RT-TM is given in Figure 4.4.

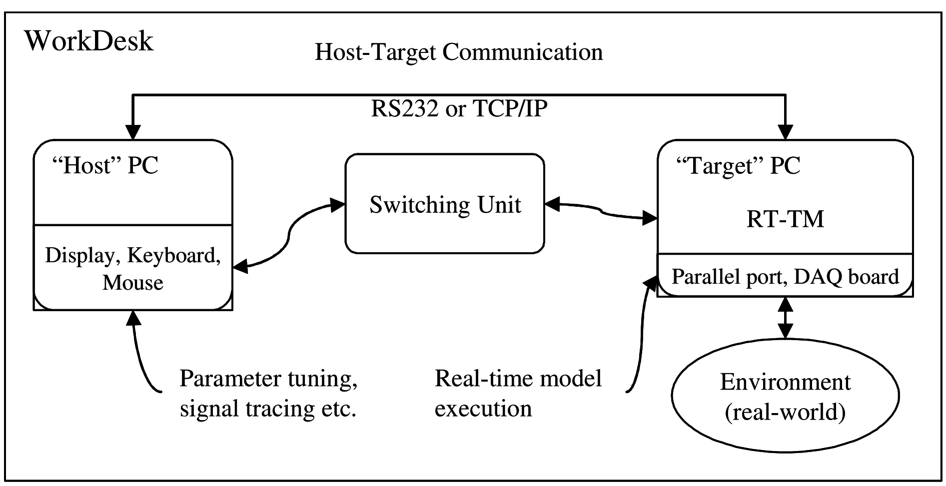


Figure 4.4 RT-TM hardware configuration

The selected software tool for accomplishing real-time code generation from models is xPC Target of The Mathworks. Real-time applications are aimed to be developed in Simulink or Stateflow on the readily available "host" computer of the WorkDesk. Thereafter, using Real-Time Workshop and xPC Target, C code corresponding to the model is generated. The generated code is then compiled using the Visual C/C++ compiler and downloaded to the Real-Time Target Module and executed. The RT-TM contains the xPC Target real-time kernel, which is loaded at the boot stage. The interaction of the RT-TM with the environment is handled using the I/O ports; i.e., parallel and serial ports, DAQ board connections.

The most attractive way of using xPC target lies on its ability to provide interactive tuning on model parameters, signal tracing and signal logging, while the model is executed in real-time on RT-TM. This is a very important advantage, which lets the developers to observe the affects of parameter changes on a real system. The "host" PC of the WorkDesk is used for this purpose, where the hard real-time processing takes on the RT-TM without interference from an operating system.

Another critical part of the RT-TM is its ability to interact with the environment. This is established through the Input-Output ports. The RT-TM can handle communication/interaction with the environment using the following ports:

- Serial Port.
- Parallel Port.
- Ethernet Port.
- DAQ Board Connections (AD, DA, DI, DO ports).

However, although the above hardware ports are readily available, the xPC Target does not have drivers associated with the Parallel Port or the DAQ Board. Actually, a very important part of RT-TM is the data acquisition board, which handles analog input/output processing. Many mechatronics laboratories do not enable the use of DAQ Boards, which generally suffer from high costs. Therefore many real-time applications cannot be practiced with the students. In order to solve for this problem, a survey has been conducted on data acquisition boards that may be used for educational purposes. The results show that there is low cost hardware, which cannot be introduced to the students as they do not have compatibility with the software development tools used, i.e., xPC Target of Mathworks. In order to overcome this conflict and to enable low cost real-time experimentation, software drivers compatible with xPC Target for the Parallel Port and the DAQ Board have been developed. This way it is intended to use low cost desktop computer hardware as low cost real-time target platforms, where available and applicable.

The general specifications of the Decision Computer International data acquisition board are given below:

- 12 bit resolution, unipolar or bipolar selectable 16 A/D channels.
- 12 bit resolution, unipolar or bipolar selectable 2 D/A channels.
- 16 bit digital input/digital output channels.
- Unipolar input/output voltage range: 0 to 10V or 0 to 20V.
- Bipolar input/output voltage range: –5V to 5V or –10V to 10V.

The software drivers developed for xPC Target are integrated into a Simulink library called the Mechatronics Education Desk Simulink Library. The library contains Simulink S-Function blocks developed in C language. This open ended library permits addition of other blocks developed in the scope of mechatronics courses, which will be united in this library so that students can distribute their work and work developed will not be lost in time.

Another inclusion to the RT-TM considered for the real-time practicing is the Embedded Target Module. The Embedded Target Module is aimed to be used as an optional component to support embedded standalone demonstration and/or project development within mechatronics courses. The E-TM components are also selected to be supported by the Mathworks xPC Target applications. The E-TM consists of PC-104 form CPU, DAQ and PSU modules. The detailed structure of E-TM is provided in the MED CD.

To sum up, the integration of the RT-TM will lead to the generation of extensible and low cost real-time experiments, which is considered to be the lowest cost system available currently. The usage of the library blocks corresponding to the RT-TM are explained in Appendix A.

4.6 Mechatronic Building Blocks

In this section, the mechatronic building blocks selected for use within the Mechatronics Education Desk are presented. The selections, remarks and general specifications for the blocks are tabulated in the coming part as actuators, sensors and microcontrollers. The proceeding section lists the compatible software modules.

Table 4.2 Mechatronic building blocks in MED

Actuators	
DC motors w/wo gear head	R/C Servomotors
DC motors w/ embedded feedback	Solenoids
DC Fans w/wo speed or temperature feedback	Peltier Modules
Step Motors	
Sensors	
Microswitches	Ultrasonic sensors
Mercury Switches	Gyroscopes
Reed Switches	Encoders
Opto-reflective sensors	Temperature Sensors
Photoresistors	Camera
Potentiometers	Pyroelectric Sensors
IR Ranger Sensors	Flame Detectors
Navigation Sensors (GPS, compass etc.)	Inductive Sensors
Microcontrollers	
PIC microcontrollers	Basic Stamp microcontrollers

In addition to these three groups software tools required are given in Table 4.3.

Table 4.3 Software modules in MED

Software Modules	**Remarks/Specifications**
Matlab Image Processing Toolbox System Identification Toolbox Control System Toolbox	Technical computing, image processing and control applications.
Simulink DSP Blockset SimMechanics Blockset SimPowerSystems Blockset	Model based design and simulation.
Software Modules	**Remarks/Specifications**
Real Time Workshop	Automatic code generation from Simulink models.
xPC Target & xPC Target Embedded Option*	Real-time code execution, external parameter tuning, signal tracing and logging.
Visual C++ Compiler	C compiler for use in software development and code generation from Simulink.
Microcontroller Software	PIC CCS Compiler and Basic Stamp Editors for microcontroller software development.
SolidWorks	CAD. An important advantage is compatibility of this software with Simulink, where students can start their design in SolidWorks and transfer the plant model to Simulink automatically.
MS Project	Project (design) planning and program making in design teams.
Electronic Workbench	Circuit development.
Marc	Finite element modeling.

*xPC Target Embedded Option is considered to be made available with a single license dedicated to the whole laboratory, to be used only at projects requiring standalone applications.

4.6.1 Interfacing Mechatronic Building Blocks

In general mechatronic building blocks in their basic form are not complete (self contained) to interface with the microcontrollers. For example, switches require pull-ups, photoresistors require voltage dividers, and microcontrollers without analog inputs need A/D converters and dc motor interfacing necessitate drivers. This in fact shows the need to have mechatronic building block interface circuitries to be developed as well. From this point of view and considering the available microcontroller boards in use, a roadmap has been determined. This roadmap includes the development of a general purpose interface board that will also be used in the autonomous mobile robot and add-on modules for task or building block specific interfacing. This way, an open interfacing architecture is aimed and consequently a versatile general use board is developed. The specifications for this board are given in the next section.

4.6.2 ready2go Interface Board

The ready2go Interface Board is low cost multi-functional interface board designed to support a processor or a controller board to facilitate interaction with a wide range of mechatronic peripherals. The board unites many hardware interface components of mechatronic building blocks into a small, easy to use and interface board package. A view of the board is presented in Figure 4.5.

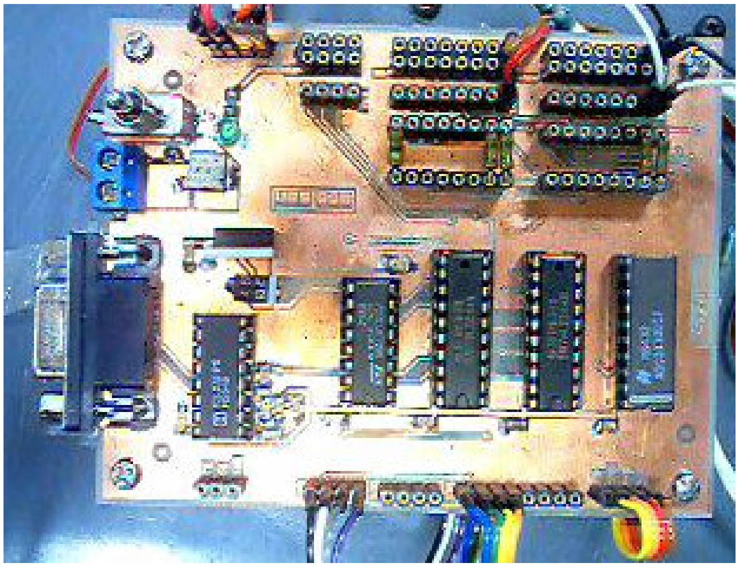


Figure 4.5 ready2go Interface Board

The specifications of the ready2go interface board are given below.

General Specifications:

- 8 analog input channels with single chip 8 bit A/D converter.
- 8 buffered digital input channels.
- H-Bridge motor driver capable of driving 4 dc motors.
- 4 buffered digital outputs.
- TTL to RS232 level converter.
- Single power supply operation and on board converter for logic supply.
- Input protection against faulty connection.
- Power on/off switch and operation indicator LED.
- DIP sockets for easy and fast replace ability of integrated circuits.
- Space for easy pull-up resistor inclusion on analog and digital input ports.

Motor Driving Capabilities:

- Driving capability of 4 dc motors at 24V/1A rating.
- H-Bridge structure for enabling direction control.

- Built in freewheeling diodes for protection from induced voltages.
- Buffered microcontroller connection for controller side protection.
- 4 buffered digital outputs with +5V and ground connections for direct R/C servomotor interfacing.

Sensor Interfacing Capabilities:

- 8 bit resolution, eight analog input channels reachable from the 4 wire microcontroller interface via synchronous serial communication.
- All analog input channels enable easy external pull-up resistor inclusion.
- Eight digital input channels.
- All digital input channels enable easy external pull-up resistor inclusion.
- All digital inputs have buffered microcontroller interface for controller side protection.

Both of the analog and digital sensor interface ports include +5V and Ground connection pins to enable direct and easy access to the sensors. There is also a MAX232 IC provided, which is used to translate RS232 level signals into TTL level or vice versa. The RS232 level signals are presented with a D-Type 9 pin standard connector as found on pc hardware and the TTL level converted signals are reachable from the controller side port connections.

The block diagram of the board is presented in Figure 4.6. The labels correspond to the connectors and inputs/outputs (I/Os) ports of the ready2go board. Further details of the board, including schematics, interfacing methods, connections and component lists are provided in the Mechatronics Education Desk Compact Disk (MED CD) under the directory named Mechatronics Education Desk/ready2go.

Due to the small size, cost and interfacing capabilities the board is also used as a part of the autonomous mobile robot kit, which will be explained in the coming part.

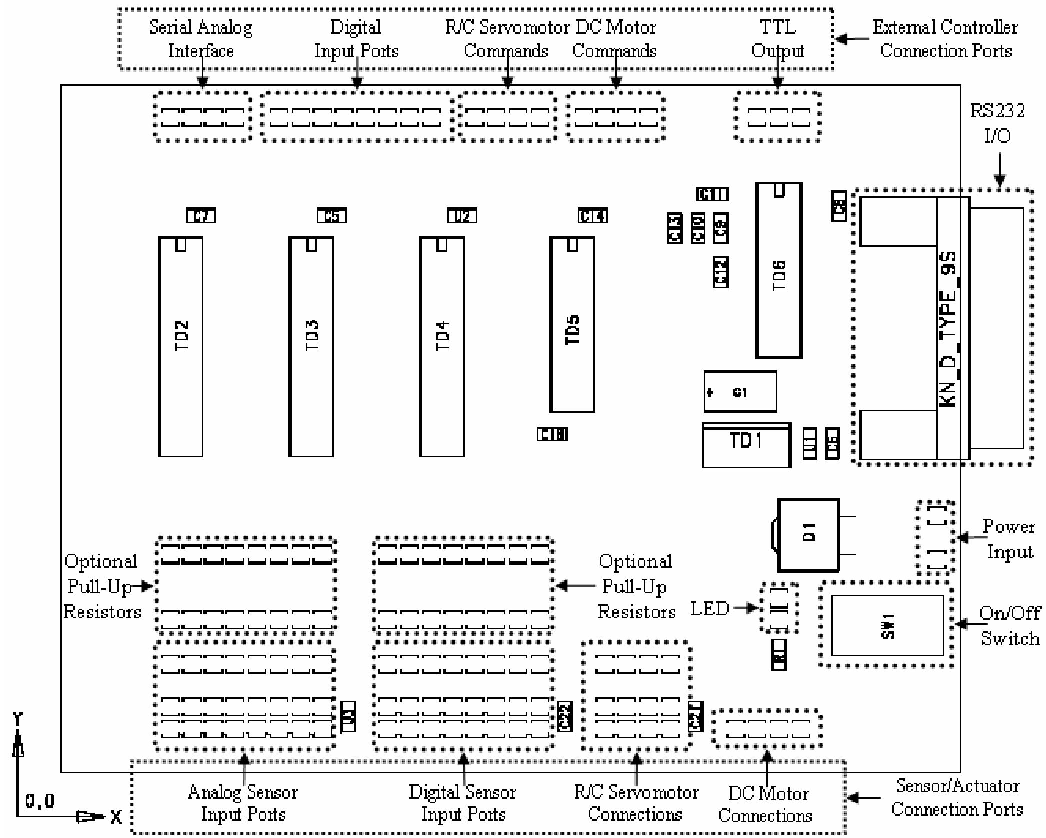


Figure 4.6 ready2go board block diagram with port labels

4.6.3 Mobile Robot Kit

Mobile robots are challenging to educators and students in many ways. First of all, working with mobile robots requires practice. It is for sure that students and even their instructors get hands-on experience through establishing a priori defined tasks through the use of mobile robots. Theory learned from lectures are turned into practice rapidly with emotion, since a result of an application even a mistake, can easily be seen in the real-world application, which is an exciting experience for everyone involved during implementation. This indeed facilitates the teaching and learning process both for the instructor and the student. Another point to be mentioned here is that the system integration and inter-disciplinary working of students. Tasks defined for the mobile robot, due its nature, cannot be limited into a single discipline and require an inter-disciplinary work and system integration. Many pieces of work from different disciplines, such as mechanics, electronics, control and software need to be integrated to fulfill a single main task defined. Depending on the application, disciplines involved are broadened and the methods applied by the students to resolve a problem get unlimited. Students learn from first hand how to integrate subsystems to form a system concept, and discuss on various aspects, such as how to assemble parts, how one subsystem interacts with the other, whether subsystems involved have an effect on one to another, and may optimization be possible for their very case, why the theory differs from practice and so on.

Another issue is that via defining tasks for the mobile robot, many mechatronic building blocks can also be taught and their usage can be experienced by the student. An important point here is the definition of a problem at the initial stage, where no limitation on the use of building blocks should be given and tasks should be defined so as to force imagination of students. This way, different outputs shall be created, where in the end through information exchange; the knowledge gained by the students can be gradually increased. It can be concluded that flexibility devoted in serving as a teaching assistant and afore mentioned values has created Mobile Robots an indispensable tool in mechatronics education.

On the light of above discussion and as a key supporting tool for the Mechatronics Education Desk and its accompanying experiments, a custom mobile robot kit has been developed. The robot is equipped with easy to use and interface equipment to provide a wide content of mechatronics practice. In its basic form, the robot kit is designed to perform the following behaviors initially: light following, wall following, line tracing and obstacle avoidance. The kit also includes chassis extensions for extra parts mounting and on-board battery for standalone operation. The architecture given in Figure 4.6 permits the application of additional robotic behaviors with the inclusion of new mechatronic building blocks.

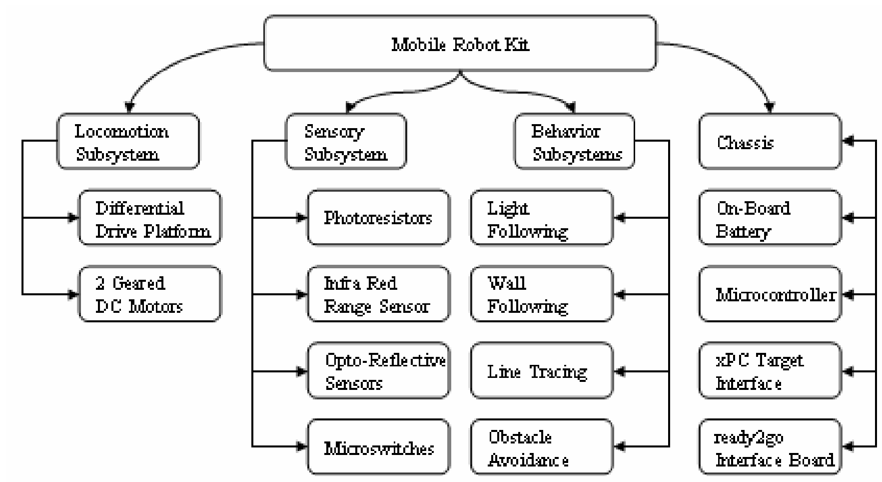


Figure 4.7 Mobile Robot Kit block diagram

In order to permit fast and reliable connection to the building blocks (drive motors, sensors and the controller) and let the students concentrate on programming for robotic behaviors and mechatronic systems integration, the ready2go interface board is also provided inside the mobile robot kit. A view of the mobile robot kit is given in Figure 4.8.

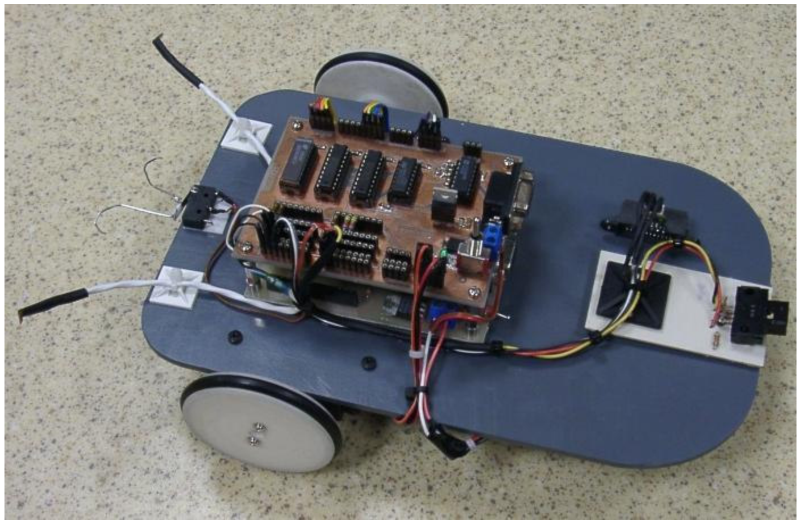


Figure 4.8 Mobile Robot Kit

General specifications of the robot are listed below:

- Three wheel differential drive locomotion.
- Two modified R/C servomotors.
- ready2go interface board.
- Microcontroller board (i.e., Basic Stamp 2 microcontroller board).
- On board Ni-Mh rechargeable battery.
- Sensor kit composed of photoresistors, microswitches, IR range sensors and opto-reflective sensors.
- Low cost easy to assemble chassis.
- Software for line-wall-light following and basic obstacle avoidance.

The coming sections address the details of the blocks represented in the architecture diagram.

Locomotion Subsystem

A three-wheeled differential drive locomotion system is selected as the means for the mobile robot to achieve its mobility within its environment. In a differential drive (steering) system the introduction of a velocity differential across the two sides of a vehicle results in a change of direction, which is the same principle applied in tracked vehicles. Consequently, the mobile robot kit has a high mobility (i.e. turning in place) enabling it a wide range of applications requiring high maneuverability.

The differential drive platform is driven by two modified Remote/Controlled (R/C) Servomotors. The reason to use modified (or hacked) R/C Servomotors lies in achieving a modular, easy to assemble, low weight and an inexpensive geared dc motor providing enough torque for the mobile robot. The ready2go board is used to interface the embedded controller or the external controller to the two modified motors.

Sensory Subsystem

Four types of sensors are made available on the mobile robot kit as a basic starting point. These sensors are 2 photoresistors, 2 microswitches, 1 Infra Red Range Sensor and 2 Infra Red opto-reflective sensors. These sensors are interfaced with the onboard or external controller via the ready2go interface board.

Behavior Subsystem

Inputs from the four types of sensors available on the kit are used to implement four common and basic robotic behaviors; light following, wall following, line tracing and obstacle avoidance.

Light Following: Light following refers to steering the mobile robot in the direction of maximum light level within the neighborhood of the vehicle. This behavior resembles actions of some insects and plants. The behavior is implemented via comparing the two photoresistors values and giving decision on the direction of the highest light level and providing this information to ready2go. This behavior can also be extended within the experiments via integration of additional photoresistors that will enable light level detection from other directions of the robot.

Wall Following: Wall following refers to steering the robot vehicle against a wall nearby from a predefined distance within the tolerances allowed. In order for the robot to fulfill its mission it is equipped with an IR Range sensor that can be used to measure distance using the triangulation technique. This behavior is useful for showing the students understand positioning of sensors on the vehicle, as they directly affect the control algorithm, and to show the affect of tolerances on smooth steering. For instance, a tight tolerance would create the robot to make higher number of steering. The sensor, targeting the wall can be directed towards the front line of the robot vehicle to avoid obstacles or follow a vehicle in front from a predefined distance. The behavioral experiments can be further extended via putting the sensor on a mini R/C servomotor that swings between certain angles to determine close obstacle locations and navigate the robot avoiding them.

Line Tracing: Line tracing refers to the act of following a line on a floor base. This is a very useful and often applied technique by Automated Guided Vehicles (AGV) in factories to navigate from one station to another. This is a very practical real world experiment that leads the students to understand the use of mobile robotics. Two IR opto-reflective sensors facing downwards in front of the robot are used to detect the position of the vehicle with respect to the line.

Obstacle Avoidance: Obstacle avoidance refers to the act of avoiding harmful contact to any obstacles found on the path. This is an important task to be completed, both for the robot itself and the obstacles encountered, in order to avoid causing harm. This may also be considered as the very basic action to become alive in the nature. Two microswitches with extensions are used to detect obstacles nearby and following a contact the robot is directed towards another obstacle free direction.

CHAPTER 5

CASE STUDIES

In this chapter case studies developed as a part of the Mechatronics Education Desk are presented. There are two experiments developed. First experiment is the Intelligent Money Selector, which focuses on the mechatronic system integration. The experiment is a well prepared demonstration of how different mechatronic building blocks are put together in achieving a main goal. The second experiment is the Heater-Cooler, which is a real-world case, a temperature control application. The importance of the experiment is in demonstrating the use of model-based design software for real-time control implementation and controller optimization via parameter tuning. The experiment is vital in showing the affect of controller parameters on a system and practicing hardware in-the loop testing.

Both experiments are designed to be in a modular fashion to let the students and the instructors modify parts of it. Another aspect is the graphical user interfaces provided with the experiments, which allows the experiments to be used at all levels of undergraduate curriculum with great ease.

Focus has been given on the use of interconnected software tools, which facilitates the experimentation and lowers the costs. The experiments carried out are good measures of how low cost equipment can be unified for mechatronics education.

The following sections contain detailed specifications for both of these case studies.

5.1 Case Study – MED Experiment 1 (Intelligent Money Selector)

5.1.1 Definition

MED Experiment 1 is named as Intelligent Money Selector (IMS). The IMS system is a sample solution to the problem of automatic selection (separation) of different coins fed by the customer. IMS, as an experiment, is designed to serve for demonstrative purposes as well as an enabling apparatus for the development and experimenting of several mechatronic building blocks and subsystems. The experiment also provides a through understanding on how bonding of these elements for achieving a single goal can be accomplished. It is mainly developed to demonstrate the “systems thinking” point of view as well as “systems interfacing” in mechatronics design.

IMS is developed in a manner to enable it to be used at all levels of mechatronics education. It is composed of modular subunits, which can be replaced by other subsystems, whenever proper interfaces are provided. The experiment, initially, can be automatically carried out aiming for technological and conceptual demonstration and thereafter can be divided into sections to teach and practice different mechatronic concepts and tools.

The brief definition of the problem for this experiment is to partition two different coin types. The solution generated for solving this problem is designed to enable demonstrating many mechatronic peripherals and unifying them in a well sequenced strategy. The solution requires; taking images of the coins, processing these images until the area of each coin is found, and acting through microcontroller driven actuators, to guide the coins to their section.

5.1.2 Aim

The below listed concepts are aimed to be introduced to the mechatronics students with the use of IMS:

- Image Processing
  - Vision Elements (i.e., Camera, Frame Grabbers, Image Types, and Image acquisition).
  - Basic Image Processing (i.e., Segmentation, Processing.).
- Feature Extraction from Images.
- Cognition/Pattern Recognition.
- Digital R/C Servomotor Control.
- Digital Step Motor Control.
- Serial (Asynchronous) Communication.
- Mechatronic System Integration and Interfacing.

The following building blocks of mechatronics are used within the IMS experiment and are aimed to be introduced to the students:

- Actuators & Drives
  - R/C Servomotors.
  - Step Motor.
  - ready2go Interface Board.
- Sensors
  - USB Camera.
- Controllers
  - Microcontroller with serial communication interface (i.e., Basic Stamp 2).
  - Personal Computer with USB port.
- Software
  - Matlab.
  - Matlab Image Processing Toolbox.
  - Basic Stamp 2 editor/compiler.
  - Microcontroller and IMS Experiment Software.

5.1.3 IMS System Architecture

The system architecture of IMS is given in Figure 5.1 below:

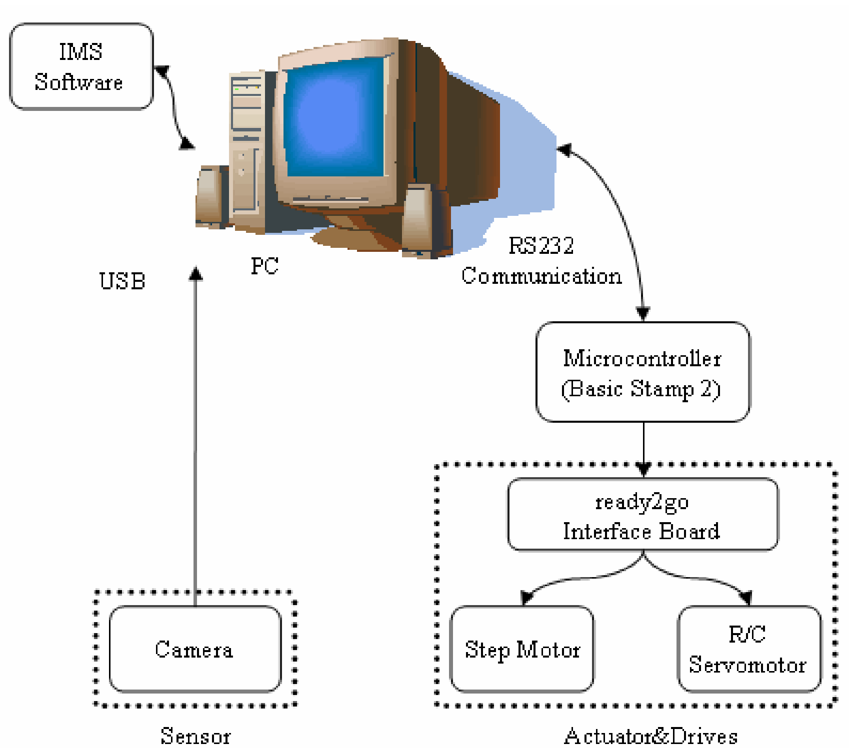


Figure 5.1 IMS system architecture

The IMS system performs the following actions, respectively:

1. Image acquisition (capturing) of the coin via a camera connected through the USB port of the personal computer.
2. Image processing algorithm application to the captured image of the coin to prepare for the cognition phase.
3. Extracting features of the coin and deciding on the action to be taken regarding the type of coin encountered.

4. Sending action (decision) information out through the RS-232 serial communication port.
5. Receiving information sent from the serial port via the microcontroller unit, which is responsible for decoding the information and acting accordingly, i.e., separate the coin to its partition with the use of actuators (stepper and R/C servomotor).

The IMS experiment is conducted through special software provided with a graphical user interface (GUI) developed. A sample view of this GUI is given in Figure 5.2.

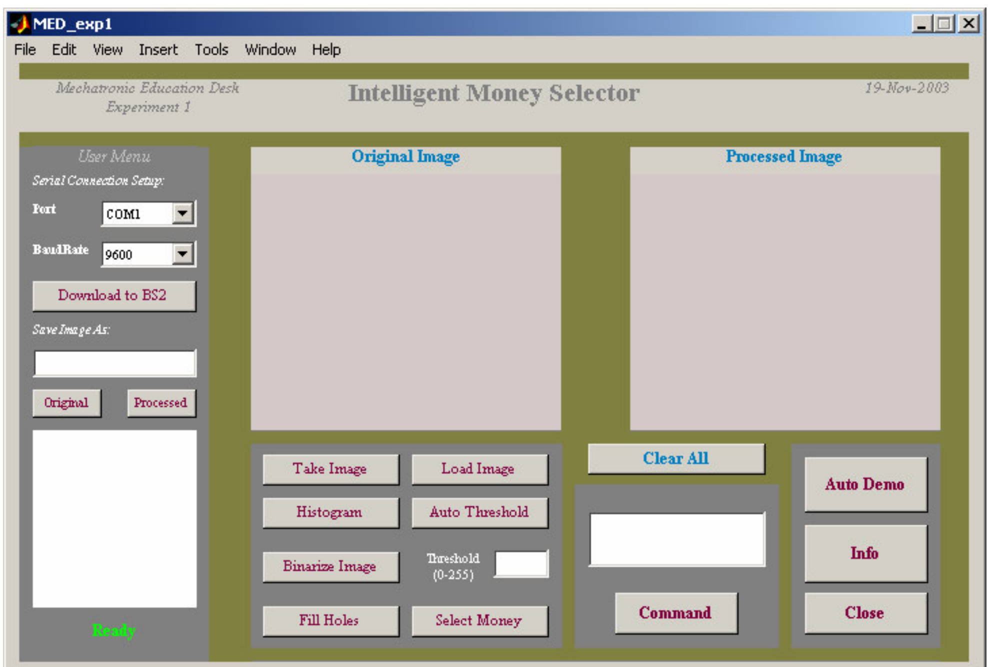


Figure 5.2 IMS experiment software GUI

The experiment software GUI allows full guidance of the experiment, which also enables the students observe and realize the affects of changes made on parameters or algorithms of the experiment.

5.1.4 Prepared Resources

This section introduces the resources, hardware and software modules prepared that will enable running the experiment. These resources include the following items:

- Physical Setup
- Microcontroller Software
- Experiment Software (GUI)

5.1.4.1 Physical Setup

The Physical Setup consists of the following listed parts:

- Camera: A Plug & Play USB camera is used to gather images of the coin. The placement and angles of the camera are made flexible for changes to take part.
- Turn-Table: A motor driven turn table is used to guide the coins to the pre-determined angles (partitioning of coins). This is the platform where the coins are placed under the camera and thereafter guided to the section they correspond to.
- Microcontroller: A Basic Stamp 2 microcontroller is used to drive the step motor and the R/C servomotor according to the communication and demands sent from the user computer via the IMS software.
- ready2go Interface Board: The ready2go board is used to drive the actuators through the microcontroller.
- Actuators: Two actuators are used to guide the coins through their section. One is a unipolar type step motor and the other is a standard R/C servomotor.

The instructions for building the setup together with the cable connections are given in the MED CD.

5.1.4.2 Microcontroller Software

Basic Stamp 2 microcontroller board developed in the METU Mechatronics Design Laboratory is used in this experiment. The microcontroller is responsible for receiving serial data from the computer, decoding the message and applying the decision via driving the actuators. The software for this purpose is prepared using the Basic Stamp windows editor. The software can be found on the MED CD.

5.1.4.3 Experiment Software (GUI)

The heart of this application is the experiment software. The software lets the students observe and realize each step of the experiment and applies the coin selection algorithm. The software is developed in Matlab as an M-File, which is also accompanied with a Graphical User Interface (GUI). The GUI enables step by step proceeding of the experiment. Installing and starting the software is explained first:

Starting Experiment Software: In order to start the experiment software the user should carry on the following instructions:

- Locate the directory of Experiment 1 on the MED CD and copy all the files to a directory on the local PC.
- Start Matlab.
- From within Matlab, locate the directory where the files are copied.
- Select and run the experiment software named: MED_Exp1.m.

Before proceeding on the details of the GUI, a brief explanation of the algorithm applied in solving this problem is given.

Coin Selection Algorithm: A sample algorithm is developed to determine the type of the coin. This algorithm is based on the fact that all types of coins are manufactured in varying diameters. That is, a unique distinction can be done when the size (i.e., area) of a coin is found. Knowing this at hand, the solution on this problem is based on determining the area of the coins and making the decision consequently. To find the area of the coin the following consecutive steps are taken:

- Image acquisition (original image),
- Gray level histogram generation,
- Determination of the binary threshold,
- Binarization of the image using a threshold value (either automatic or manual),
- Filling the holes in the generated binary image, so that an area of the coin can be extracted and finally,
- Determination of the area of the object available on the binarized and filled image (object).

All of these steps defined can be manually applied by the student using individual buttons dedicated. Therefore, the affect of each process on the image can be observed easily. The details of these steps are explained in the next part, under the property of each dedicated button.

The coming section introduces the features of this software in brief. A screenshot containing numbered labels of different features of the GUI is given in the next figure. The labels and their corresponding properties are also explained below.

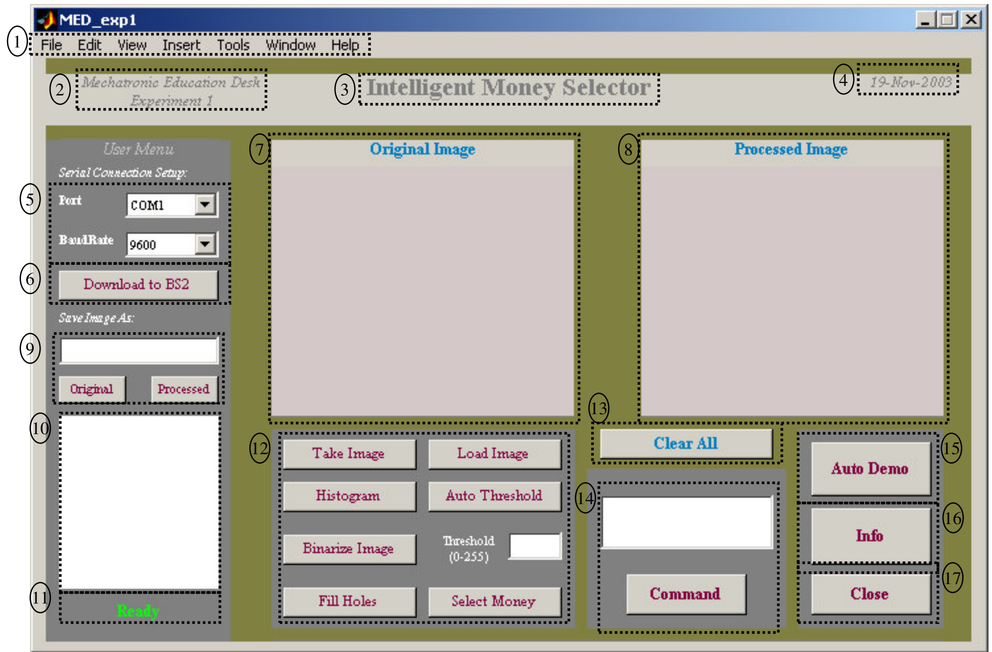


Figure 5.3 IMS experiment GUI with labels

1. **Figure Menu:** In this section a Figure Menu is provided. This menu is an option served with Matlab, which enables executing of file operations (File Open, File Save, Printing, etc.), editing figure properties (zooming, selecting etc.), inserting labels and more.

2. **Experiment Label:** Label for the Mechatronics Education Desk and Experiment 1 are presented.

3. **Experiment Name:** The name of the experiment is given.

4. **Date:** The current date of the computer is automatically written in this section.

5. **Serial Connection Setup:** Before starting the experiment, setup for the serial port to be used for communicating with the microcontroller should be set. The Port section corresponds to the communication port (COM1, 2, 3, 4)

where the microcontroller serial port is connected and the Baud Rate (9600, 19200…) is the communication speed set with the microcontroller. The Baud Rates should be set equal both on the microcontroller and the PC side.

6. **Download to BS2:** When this button is activated, an executable as shown in Figure 5.4 will come to focus after running a previously built file. This executable is prepared to download the microcontroller software to the Basic Stamp 2 board. The instructions supplied by this executable should be taken in order to successfully download the software to the target controller.

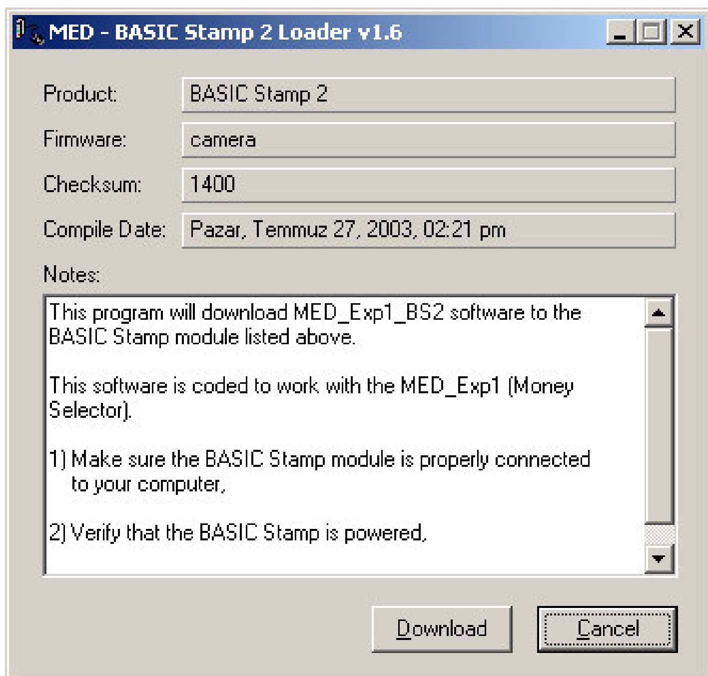


Figure 5.4 IMS experiment microcontroller software downloader

7. **Original Image:** This is the axes where images taken via the external camera or previously taken original images are shown. The name Original refers to

"as is", that is the images loaded to this section are preserved until they are cleared or the program is closed.

8. **Processed Image:** This is the axes where processed images are shown. Original images, after any consecutive process are named as Processed Image and are shown in this section. The two axes serve for showing how changes are taking place.

9. **Save Image As Area:** This area lets the user to save the Original and Processed Images in the image type entered as text in the editor area. The editor area is the place where the user wishes to give the name for the saved image. The buttons Original and Processed, when activated saves the Image located in the corresponding axes with the name entered in the editor area. For instance, entering "a.jpg" into the Save Image As editor area and activating the Processed button, saves the Processed Image axes as an image with the name "a.jpg". Thereafter, this image can be used later for further processing by the user. This feature is quite essential in taking real-time images from the camera and saving these for later offline processing.

10. **Detailed Status Bar:** All actions or processes executed from within the GUI are written in this Status Bar to let the user have a log of the steps taken.

11. **Status Bar:** This area is used to inform the user from the current state of the software. The user is guided or informed from this section on the actions to be taken. For instance, a label "Ready" informs the user that the software is waiting for the user to make an action. On the contrary, the label "Auto Thresholding" informs the user that a process (Automatic thresholding) is being executed and the user should wait before executing a new command until the software finishes its current job.

12. **Execution Menu:** This section contains main buttons for activating the steps in performing the experiment. The properties of these buttons are also presented below.

- *Take Image:* When this button is activated, a real-time image is taken from the camera attached to the system (provided that the camera is on) and is shown on the Original Image axes. Capturing of the image is made available by the use of the USB camera Windows driver and a freeware software called VFM (Video for MATLAB) [79], which calls the windows driver routines from within Matlab. Each time the button is pressed the same action is carried on. The image taken is also transferred into a matrix variable in Matlab for further easy processing and enabling access to the user.

- *Load Image:* When this button is activated a new menu (dialog box) as shown in Figure 5.5 is shown to the user, to make a selection of the image file to be loaded to the software. After the file is selected, the image is loaded to a matrix variable and also shown to the user from the Processed Image axes.

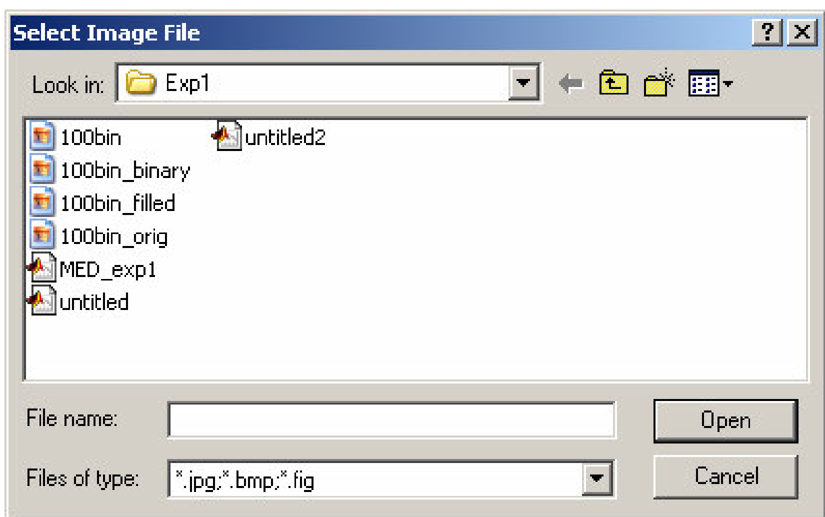


Figure 5.5 Image file selection dialog box

- *Histogram:* When this button is activated and if an Original Image is available, the gray level histogram of the Original Image is generated and shown on a new figure window.

- *Auto Threshold:* A demonstrative automatic thresholding algorithm is implemented to determine the binary image that will be used to separate the coin from its surroundings, i.e., the background, so that the shape of the coin can be realized. There are several thresholding methods that can be applied for this purpose. A look at the histograms generated show a clue on the case, which is best shown in figure 5.6. As expected, since there is a single object in the picture, there are two areas of interest in which the dark area corresponds to the object (coin) and the second to the surroundings. So, finding a single threshold value that can separate these areas can make the coin realized.

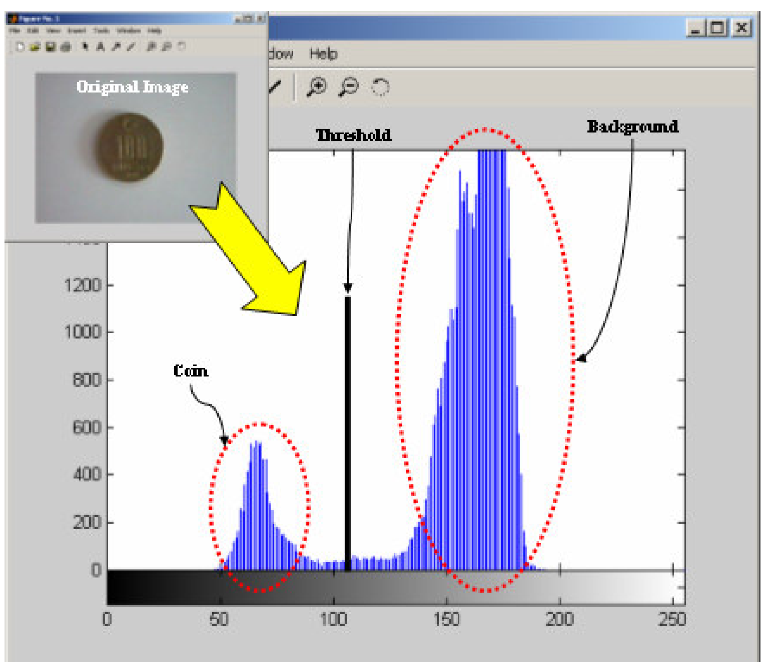


Figure 5.6 Coin image histogram

In order to find the threshold an Iterative Automatic Thresholding algorithm is implemented. This algorithm works on the fact that the image is likely to have two areas of interest. Therefore, initially the mean of the gray level histogram value is found and the threshold is assigned to take this value. Then the iteration process begins, which will divide the picture into two halves from the Threshold values that will be updated in every iteration step. In the iteration procedure, each pixel of the picture is compared to the threshold value and the image is divided into two areas, with respect to the threshold. After searching the full image and forming two areas, the mean of each individual area is calculated. Thereafter, these two mean values are summed up and the half of the value is assigned to be the new threshold value, which gets better with each iteration step. The iteration then continues to a new comparison until the iteration number defined previously is reached. In this experiment, two iterations are used to determine the threshold value. When the Auto Threshold button is activated, the above mentioned algorithm is run and a Threshold value is assigned and shown to the user.

- *Binarize Image:* When activated, the original image is binarized with respect to a threshold value. The threshold value can be either the Automatic Threshold value estimated in Auto Threshold phase or a value entered on the threshold editor area. The GUI automatically checks whether there is a valid (a value between 0 and 255) threshold value entered and gives priority to use this value. In the case when there is no threshold entered manually, the software uses the Automatic threshold value.

- *Fill Holes:* When this button is executed, a simple algorithm is run to determine and fill the holes existing in the image. The holes are defined as a set of background pixels that cannot be reached by filling in the background from the edge of the image. Filling holes is required to determine the area of the coin correctly, which removes the holes generated due the binarization process. For instance, the background (white pixels) lying inside the coin is treated as holes as shown in Figure 5.7.

Figure 5.7 Holes in the binarized coin image

- *Select Money:* When this button is activated, the image area for the object (coin) is computed. The computed area is then compared to the readily available area data and the coin type is selected accordingly. After determining the coin type, the serial communication port is opened via the setup parameters selected in the Serial Communication Setup area and the data corresponding to the coin type is sent out. The result is also shown to the user for information. With the execution of this step, the implementation stage for the GUI is terminated where the rest of the experiment is handled from the microcontroller that will partition the coins accordingly.

13. **Clear All:** When this button is activated, the memory of the software is cleared. The images loaded as well as variables corresponding to thresholds are all reset to zero. The status bar as well as any other information is also cleared. This indeed, sets the software to its initial form.

14. **Command Area:** This area is intended to enable the students enter Matlab commands as well as execute their own codes from within the GUI. The commands entered in the editor area are executed when the Command button is pressed.

15. **AutoDemo:** The execution of this button causes the software to run automatically, beginning from the image acquisition phase to the final coin separation.

16. **Info:** An information document is opened when this button is activated. The user can have information regarding to the experiment using the document served.

17. **Close:** The close button closes the experiment software and removes any memory allocation.

In the coming part, sample roadmap for executing the experiment is given.

5.1.5 Executing IMS Experiment

The IMS experiment may be run in the following order to present the previously mentioned mechatronic concepts and tools.

Phase 1 – AutoDemo Phase

Phase 1 of the experiment covers the AutoDemo action, which conducts the complete experiment automatically with the predefined parameters, algorithms and tools prepared. The main goals of this action is to provide the students with the overall “picture” of the experiment clarify the aims, define the problem, and make a clear understanding of what shall be taught and where the taught material can be used. This way, it is intended to decrease the learning period of the students and increase efficiency of the experiment via creating enthusiasm as well as focusing on the subject matter.

The AutoDemo action starts with the image acquisition stage, where one of the coins available of the predefined types is located under the camera setup and an image of this coin is captured via the USB camera. The captured image is then shown to the user on the Original Image area of the GUI and loaded to a matrix variable for further handling within the software. The next step involves plotting of the gray level histogram and showing this in a permeate figure. Following this, the Iterative Thresholding algorithm is run to determine the threshold value. After determining the threshold, the binarization of the image with respect to this threshold value is done. The original captured image together with the binarized image is shown in Figure 5.8.

Figure 5.8 Original coin image and binarized coin image (Threshold = 123)

The binarized image is also shown on the Processed Image area of the GUI. This shows the differences and affects caused in every proceeding step of the experiment. Following this stage, the binarized image of the coin is further enhanced to let easy differentiation between other coin types available. This is made possible via filling in the “holes” of the coin. This permits the calculation of the total area of the coin over the full image area. Figure 5.9 shows the processed binarized coin image together with the coin image with filled holes.

Figure 5.9 Binary coin image vs. coin image with holes filled

The newly processed image containing filled holes is again shown to the user. Afterwards the area corresponding to the coin is calculated and shown on the GUI. In the next phase, the coin type is determined from the comparison of the readily available coin area values and the result is displayed within the GUI.

As the cognition phase is completed, the next step involves the generated decision information to be transferred to the microcontroller that is responsible for the actuating mechanism, which will eventually partition the coin accordingly. Prior to the communication phase, the previously prepared microcontroller software is downloaded to the target microcontroller. This step plays a key role on showing how an embedded system is targeted and software downloaded.

Following the software download stage, the selection of the coin is sent out through the serial communication port. A message containing the coin selection is sent using RS-232 asynchronous communication port. This message is then received by the microcontroller, which decodes this message in order to determine the action to be taken. Depending on the gathered data the microcontroller then drives the RC servomotor and the step motor and via the mechanisms involved partitions the coin, which eventually completes the experiment.

All through the experiment phases, every action taken is displayed and corresponding messages are made available within the software GUI provided. A sample figure regarding this information is shown in Figure 5.10.

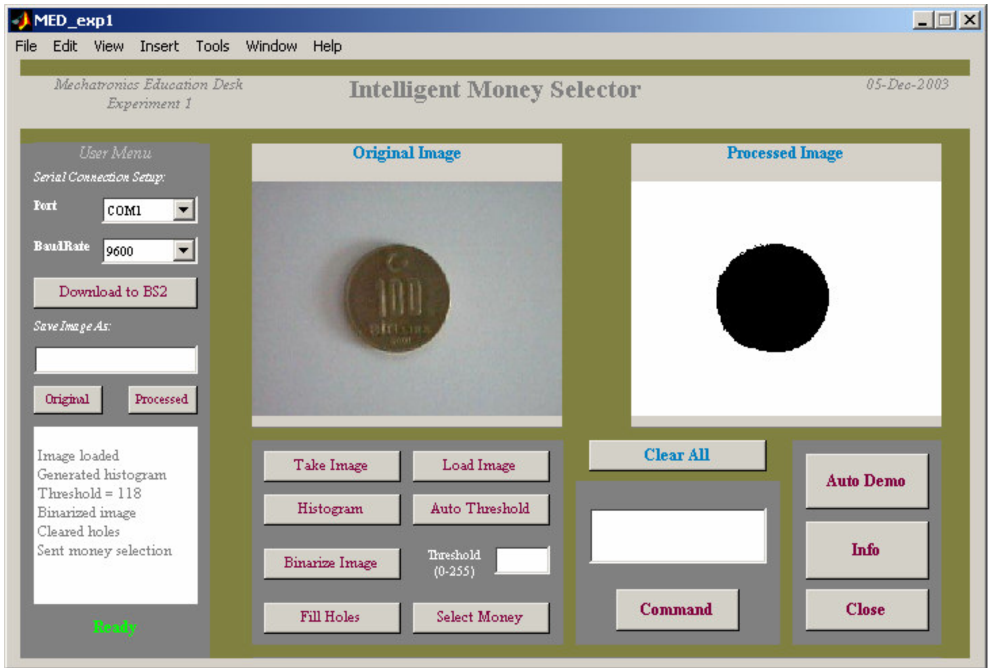


Figure 5.10 A sample view from the GUI after execution

Phase 2 – Interaction Phase

The Interaction Phase involves hands-on experimenting of each stage of the AutoDemo Phase, where every step can be carried out by the student, individually. Also, modules within the experiment can be altered or added and their corresponding affects can be observed and identified.

5.2 Case Study – MED Experiment 2 (Heater-Cooler)

5.2.1 Definition

MED Experiment 2 is named as Heater-Cooler (H/C). The Heater-Cooler experiment is an example of the very commonly encountered application of temperature control. This experiment is essential in demonstrating real-world applications from within the industry that challenge many engineers. The experiment resembles the temperature control applied in most processors and is a good example of experimentation and measurement efforts carried out in the design stage of conduction and convection cooled electronic equipment. A view of this experiment is given in Figure 5.11.

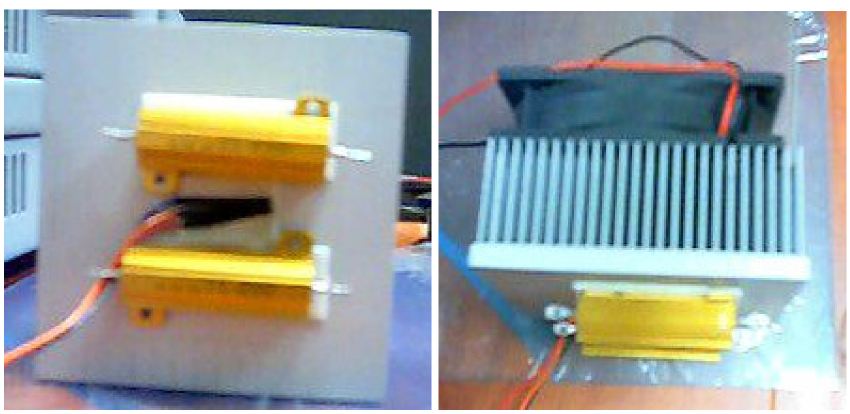


Figure 5.11 A view of the Heater-Cooler experiment

The main task defined herein is to control the temperature of a heated surface. The heat source is a two level selectable constant heat source compromised of high power resistors. Cooling is achieved with a fin attached on the heated surface with a dc fan located above, which creates forced convection cooling affect via air flow. A temperature sensor located on the center of the fin is used to measure the temperature of the surface being heated. This configuration associated with power resistors as heating elements are shown in Figure 5.12.

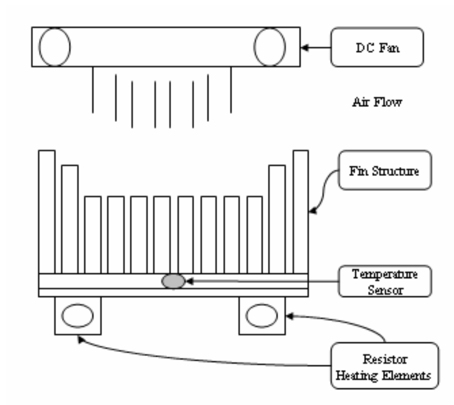


Figure 5.12 Experiment configuration for H/C

H/C experiment is designed to serve for technology demonstrative purposes, especially on the use of model-based design software with an application to a sample real-time control system. The most important contribution of H/C experiment lies in the resources, hardware and software modules that are brought together that enable easy and cost effective interfaces for “Model Based Design” and “Real-Time Control”. The experiment uses the RT-TM hardware with the Simulink Libraries developed for the data acquisition board.

H/C is also developed in a manner to enable it to be used at all levels of mechatronic education. The trainees can easily be introduced to the details of subjects on demand of the instructor. The modularity presented within the experiment also enables the created resources for this specific experiment to be transferred into other applications.

The experiment is supplied with custom developed software with an easy to use Graphical User Interface. The experiment software provides two stages of experimentation. The first is the Auto Demo stage where the experiment is carried on automatically and autonomously and the second phase corresponds to individual working of the trainee with the experiment software and models provided.

5.2.2 Aim

The below listed concepts are aimed to be introduced to the mechatronics students when conducting the Heater-Cooler experiment:

- Measurement & Data Acquisition
- Control Strategies
- Temperature Control
- System Modeling
- Digital control of DC Fans
- Pulse Width Modulation (PWM)
- Real-Time Control
- Targeting Real-Time Systems
- Using Model Based Design Software.

The following mechatronic building blocks are used in this experiment and are intended to be introduced to the students:

- Actuators & Drives
  - DC Fan
  - ready2go Interface Board.
- Sensors
  - Temperature Sensor w/ Analog Output.
- Controllers
  - Real-Time Target Module with PCI Data Acquisition Board.
- Software
  - Matlab/Simulink.

- Real Time Workshop with xPC Target option.
- Experiment 2 Software, which is special control software with Graphical User Interface and Simulink control model developed for operation with the Real-Time Target Module.

5.2.3 H/C System Architecture

The system architecture of H/C is shown in the figure below:

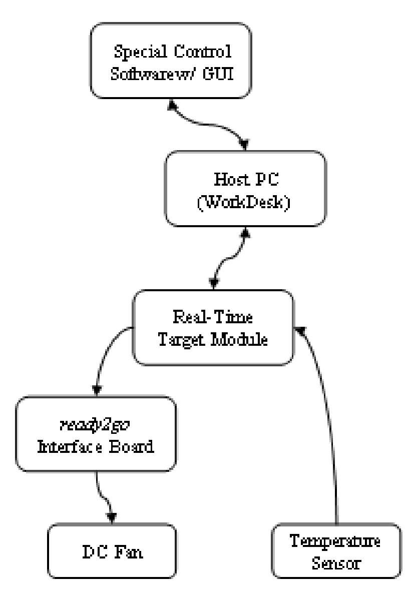


Figure 5.13 H/C system architecture

The H/C experimental system is used to perform the following actions, respectively:

1. Creating connection with the Real-Time Target Module and downloading of real-time control model to the target.
2. Starting of real-time code on the RT-TM.

3. Switching on the heater and online observing of temperature change on the heated surface remotely from the host computer.
4. Application of two selectable control strategies: On/Off and PI control, respectively to set the surface temperature to a desired level.
5. Parameter tuning (updating) of controllers, signal tracing and signal logging while the controller model is running in real-time on the RT-TM. Observation of affects of parameter updates on controllers.
6. Stopping real-time code and plotting of the real-time collected data on the host computer for evaluation of the affect of different controller strategies.

The H/C experiment is conducted through a special software provided with a graphical user interface (GUI) developed. A sample view of the GUI is shown in Figure 5.14.

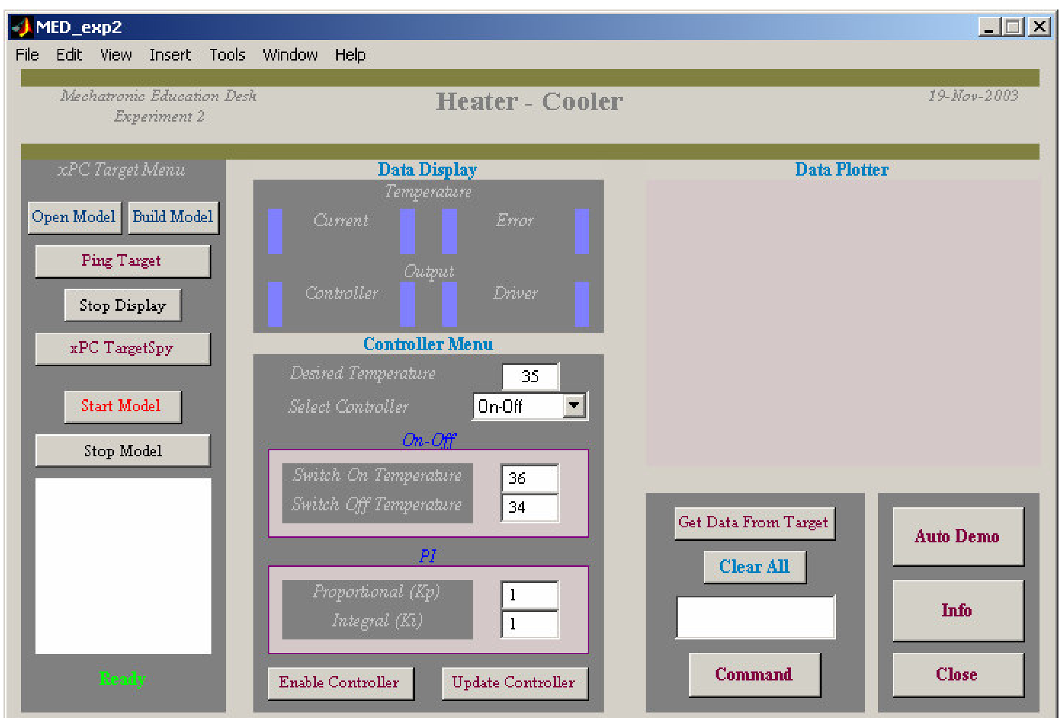


Figure 5.14 H/C experiment software GUI

The experiment software GUI allows a full guidance of the experiment to the user. The key point of the H/C experiment lies in its system architecture, especially on the software used. The software allows the following to be accomplished while the system is running in real-time:

- Parameter Updating
- Controller Tuning
- Signal Tracing
- Signal Logging

5.2.4 Prepared Resources

This section introduces the resources, hardware and software modules prepared that will enable running the experiment. These resources include the following items:

- Physical Setup
- Simulink Controller Model
- Experiment Software (GUI)

5.2.4.1 Physical Setup

A simple setup is constructed to conduct the experiment. The physical setup consists of the following listed parts. A view from the physical setup is also given below:

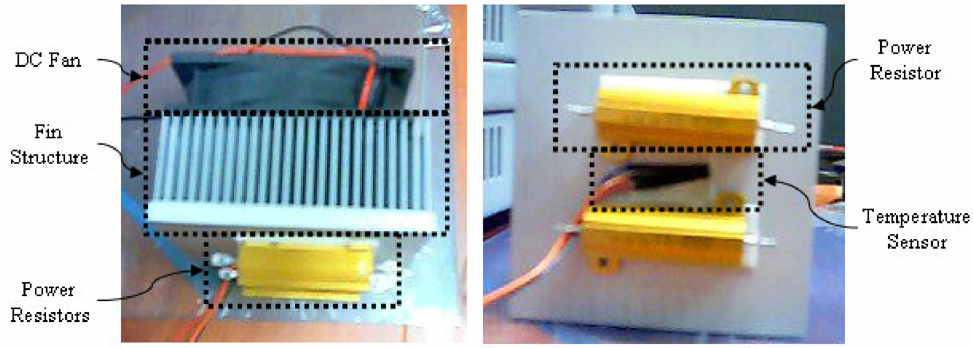


Figure 5.15 Experiment 2 physical setup

1. Heating Elements: Two high power resistors (selectable), directly connected to an external power supply, provide necessary heating effect on the fin surface.
2. Fin Structure: The fin structure, which contains sections reserved for temperature sensor and heating element mounting, is where the surface temperature control is applied.
3. Mechatronic Building Blocks: A DC fan, a temperature sensor (LM35) with analog output, ready2go Interface Board and RT-TM are used in this experiment.

The instructions for building the setup together with the cable connections are given in the MED CD.

5.2.4.2 Simulink Controller Model

Before proceeding into the details of the software models developed, the scenario of the experiment will be briefly explained.

The experiment requires the temperature control of the surface of a fin, which indeed necessitates the formation of a closed loop control system. This system can be depicted as shown in Figure 5.16:

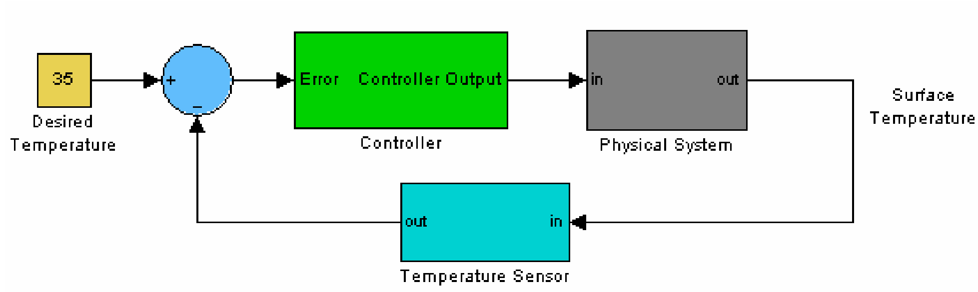


Figure 5.16 Closed loop temperature control system diagram

The functional flow is straightforward; a desired temperature requirement is given and subtracted from the actual temperature to find the error associated, which is fed to a controller where the output is applied to the actuator (actuator driver) that eventually acts to the environment.

As the aim of the experiment is considered, the above figure is transferred to a Simulink model that will be later run in real-time on the RT-TM module and will be in contact with the real-hardware (i.e., temperature sensor, dc fan). This way, hardware in-the loop (real plant) simulation setup will be established and can be used to control the system. The associated Simulink model created for this purpose is presented in Figure 5.17. The model is formed to be compatible with the xPC Target option; therefore it contains some blocks specific to this option.

The functional flow using this diagram is as following. A desired temperature is entered to the system in centigrade (Desired Temperature Block). This value is then passed through a function (T2V Block), which converts it to Volts, to the same unit and scale of the temperature sensor. The current temperature is available via the Data Acquisition Board, which is directly connected to the temperature sensor. The sensor signal is received by the readily available Simulink block named Temperature Sensor. Afterwards, the two voltage levels, desired temperature and current temperature, are subtracted and the resulting error signal is fed to the two controllers. The subsystems of the controller blocks are presented in figures 5.18 and 5.19.

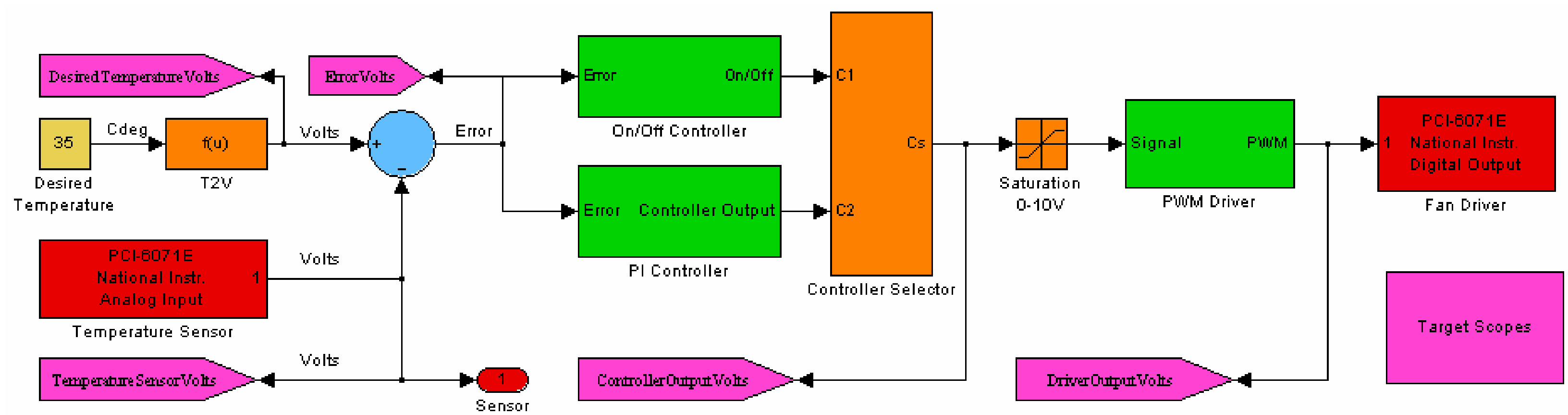


Figure 5.17 Experiment 2 Simulink model

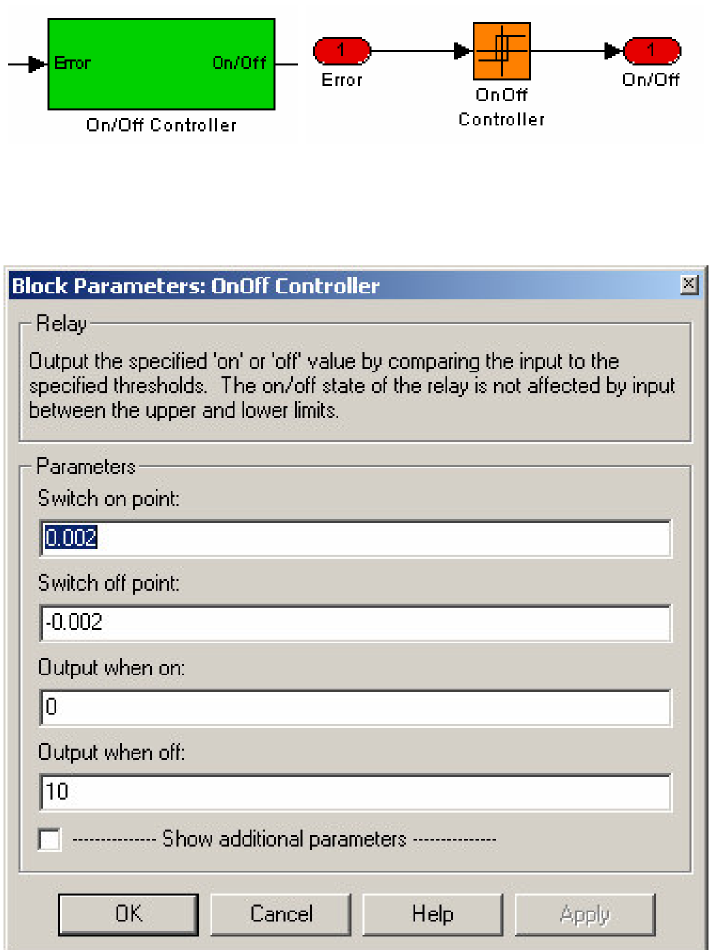
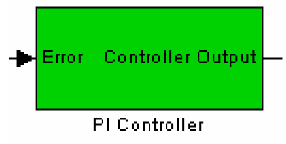


Figure 5.18 On/off controller Simulink blocks



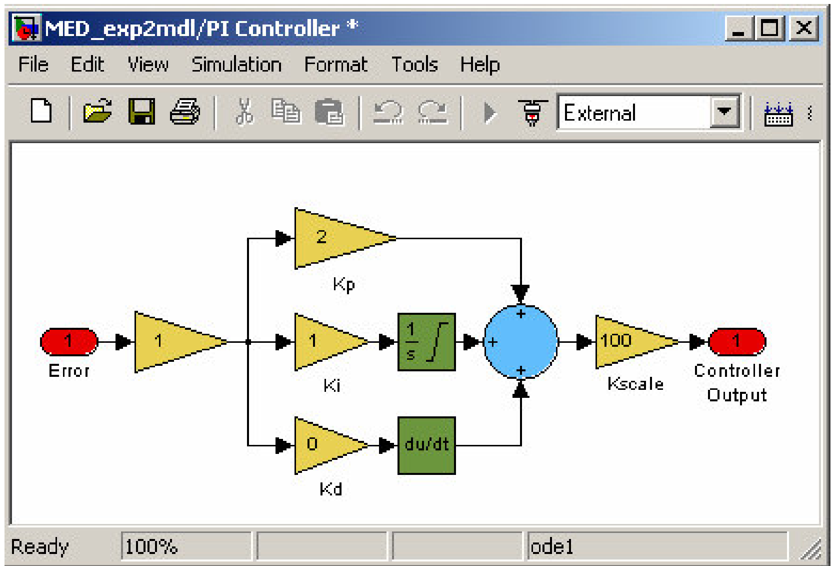


Figure 5.19 PI controller Simulink blocks

The On/Off Controller block is a simple application of a relay, which has switch on and off points with settable output values corresponding to the relay state. For instance, for the figure shown, the switch on point 0.002 volts corresponds to 2 mVs, which is 0.2 centigrade when scaled to the sensor output (The LM35 sensor output is proportional with temperature in centigrade, where 1 centigrade change causes 10mVs of output signal). So, whenever the On/Off controller receives an error over 2mVs, it will output 0Vs and 10Vs otherwise. A 0V output causes the actuator (dc fan) to stop and 10Vs causes the fan to work in maximum speed.

On the other hand, the PI Controller computes a proportional and integral term from the errors, which are then summed up and multiplied with a constant gain. The gain S_{cale} is used to scale the controller output in mVs to the fan driver input in Vs.

The outputs of the controllers then pass through the Controller Selector. This block, shown in Figure 5.20 is used to activate the desired control strategy or disable the controller via providing 0 constant outputs.

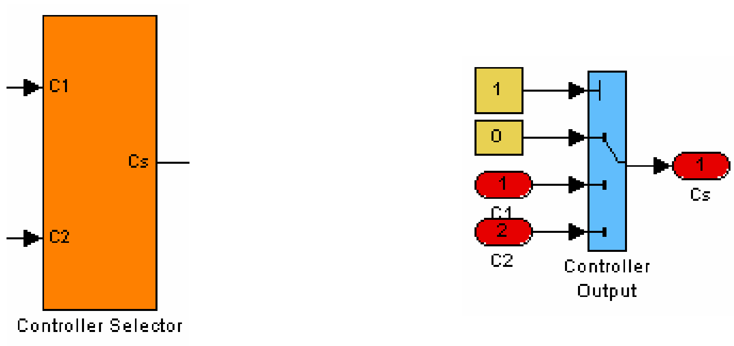


Figure 5.20 Controller selector block

The working principle of the Controller Selector Block is quite simple. The Controller Output is connected to the line, which is entered in the first constant block (Constant = 1 for the figure above). So, a value of 1 in the constant block causes the output to be 0, whereas a value of 3 causes the output to be connected to C2, which is the PI controller output.

The outputs of the controllers are later fed to a saturation block that guarantees the level supplied to the driver is between 0-10Vs. The details of this block are also given below.

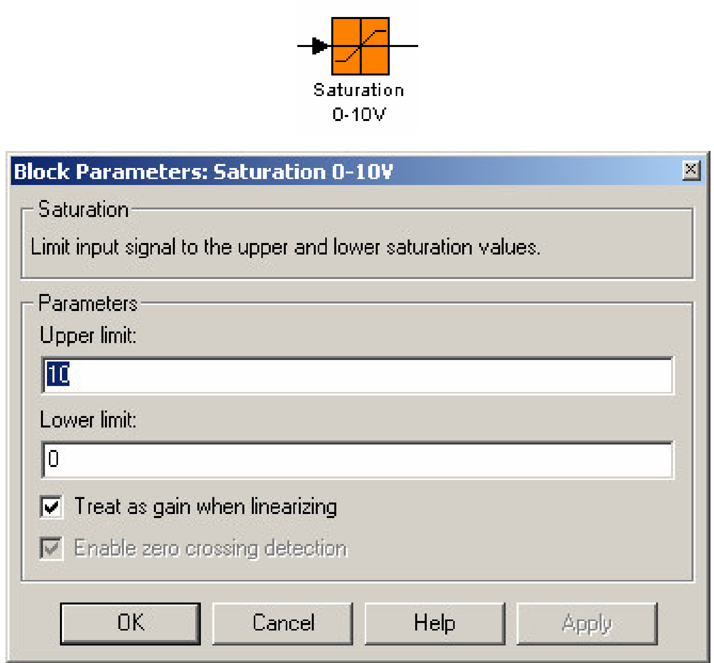


Figure 5.21 Controller output saturation block

The signals are then fed to a PWM driver block, which converts the analog signal inputs to PWM pulses. For the On/Off controller this block has no use, since the output will be in a constant level, either 0 or 10Vs. However, the PI controller requires the application of varying speeds on the actuator side, which will depend on the error associated. The PWM block computes the duty cycle required and outputs a series of pulses that will eventually resemble an analog output.

The PWM driver output is linked to the Data Acquisition Board Digital Output, which is connected to the motor driver input on the ready2go interface board. As the DC motor is connected to the accompanying motor output on the board, a pulse output of 1 will cause the dc motor to turn and a 0 pulse will make it stop.

The Simulink model constructed is used by the experiment software to conduct the experiment. The following section explains the details on how this is accomplished, presenting the structure of the experiment software.

5.2.4.3 Experiment Software (GUI)

Experiment 2, similar with experiment 1, is also run using the experiment software with a GUI developed. This point forward will focus on this software.

The experiment software is developed in Matlab as an M-File, which is also accompanied with a Graphical User Interface (GUI). The GUI enables step by step proceeding of the experiment. Simulink models, mentioned in the previous section are also run on the target module from within this software. Installing and starting the software is explained first:

Starting Experiment Software: In order to start the experiment software the user should carry on the following instructions:

- Locate the directory of Experiment 2 on the MED CD and copy all the files to a directory on the local PC.
- Start Matlab.
- From within Matlab, locate the directory where the files are copied.
- Select and run the experiment software named: MED_Exp2.m.

The coming section introduces the features of this software in brief. A screenshot containing numbered labels of different features of the GUI is given in the next figure. The labels and their corresponding properties are also explained below.

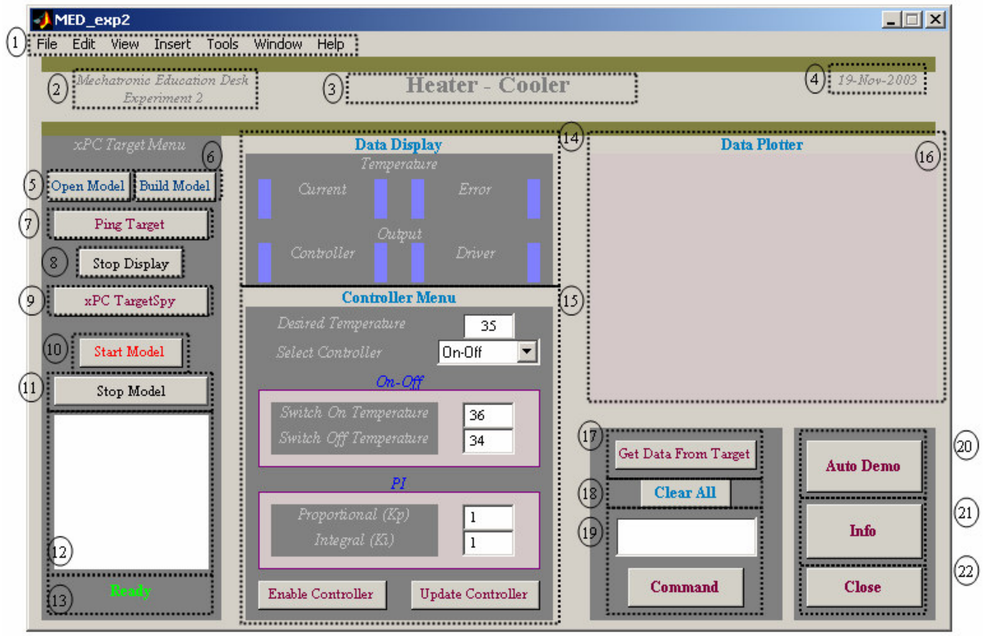


Figure 5.22 Heater-Cooler experiment GUI with labels

18. **Figure Menu:** In this section a Figure Menu is provided. This menu is an option served with Matlab, which enables executing of file operations (File Open, File Save, Printing, etc.), editing figure properties (zooming, selecting etc.), inserting labels and more.

19. **Experiment Label:** Label for the Mechatronics Education Desk and Experiment 1 are presented.

20. **Experiment Name:** The name of the experiment is given.

21. **Date:** The current date of the computer is automatically written in this section.

22. **Open Model:** When activated a window dialog box shown in Figure 5.23 is opened and the user is asked to select a Simulink model file to open it. This button can be used to open and analyze the Simulink model for the H/C experiment.

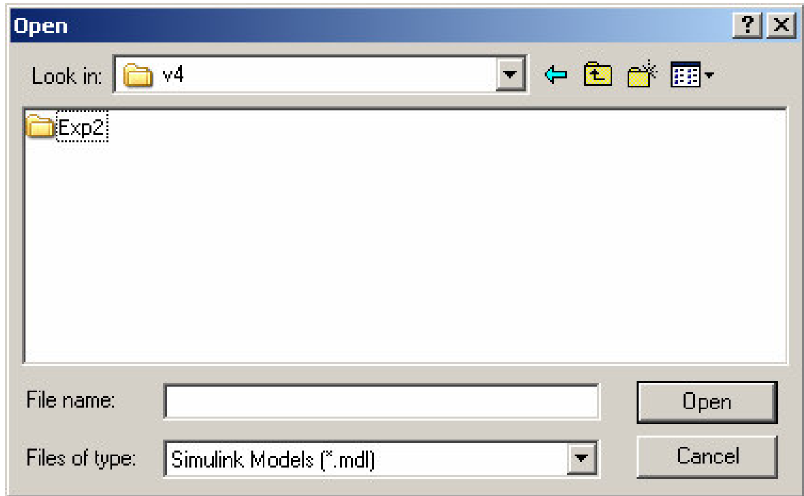


Figure 5.23 Open Simulink model dialog box

23. **Build Model:** When this button is pressed the following two actions are taken respectively:

    - The connection with the Real-Time Target Module is checked. A ping action is taken and if the result is all right, that is if the target module is open and communicating with the host computer the software proceeds to the next step. Otherwise, a warning dialog is shown to the user to indicate that there is no connection with the target.

    - After a valid connection, the software builds the Simulink Controller model and automatically downloads it to the target. The build process covers automatic C-code generation from the Simulink model, creating a real-time executable and downloading of the executable to the target module.

24. **Ping Target:** This button enables the user to check the connection with the real-time target module whenever activated. The result of a ping process is returned to the user for information.

25. **Stop/Start Display:** When code is executing on the target module a communication link is automatically generated, which continuously receives the following signal values: Temperature, Temperature Error, Controller Output and Output of Driver. The received values are automatically shown on the Data Display Menu and the Data Plotter axes. The plotting of this data can be enabled or disabled when this button is activated. It should be noted that for parameter updating on the target module, this option should be set to Disabled to stop signal tracing and start parameter updating.

26. **xPC TargetSpy:** When activated the current screen shot of the target module is shown on the host computer screen.

27. **Start Model:** This button starts the execution of the real-time executable code corresponding to the Simulink Controller Model of the experiment on the target module. Provided that the Physical Setup is active, this step enables the application of closed loop control.

28. **Stop Model:** When pressed this action stops the execution of the real-time execution of the target module.

29. **Detailed Status Bar:** All actions or processes executed from within the GUI are written in this Status Bar to let the user have a log of the steps taken.

30. **Status Bar:** This area is used to inform the user from the current state of the software. The user is guided or informed from this section on the actions to be taken.

31. **Data Display Menu:** In this menu, four display areas corresponding to the following signals are provided: Current Temperature, Temperature Error, Controller Output and Output of the Driver. Data is only shown when the

real-time executable is running and the Start Display button is active, which enables communication with the target model.

32. **Controller Menu:** This is the section where the controls for the controllers are provided. The dedicated areas are explained in the following:

- Desired Temperature: The required temperature of the surface is entered to this area as centigrade. The value written here is sent to the target module for parameter updating, when the Enable Controller button is active and the Update Controller is pressed once after entering the value.

- Select Controller: This popup menu provides two control strategy selections: On/Off and PI control. The selected controller type is activated.

- Switch On Temperature: This editor area is provided to set the Switch On Temperature (centigrade) for the On/Off Controller.

- Switch Off Temperature: This editor area is provided to set the Switch Off Temperature (centigrade) for the On/Off Controller.

- Proportional (Kp): This editor area is provided to set the value of the Proportional gain value on the PI controller.

- Integral (Ki): This editor area is provided to set the value of the Integral gain value on the PI controller.

- Enable Controller: The button is provided to enable/disable the controller. When disabled, the Controller Selector value on the Simulink Model is given a value of 1, which outputs a constant 0Vs that disables the actuator. In this mode, temperature sensor values can still be observed and logged. This mode is especially useful to determine the heating characteristic (disturbance) of the fin structure due to the constant heat sources activated. When enabled, the controller type selected from the Select Controller popup menu is activated.

- Update Controller: A feature of the software is that it does not enable the user to change parameters on the target module without the user demand. Any value change made on the Controller Menu is only activated when the real-time code is executing and the Update Controller button is pressed. If values entered in the Controller Menu are not valid data or the Update Controller button is not pressed default values used in the Simulink Model are active.

33. **Data Plotter Area:** In this section, when the model is executing on the target module, the surface temperature received from the target is plotted in centigrade.

34. **Get Data From Target:** Signals can be logged in real-time on the target module. In default for this experiment, temperature sensor output is logged continuously with respect to time. The data are logged on the Ram of the target module and after execution can be transferred to the host computer for evaluation via activating this button. The data received is automatically plotted on the Data Plotter axes. A sample data logged is shown in Figure 5.24. During this log, the heating elements were on and no control was applied, the dc fan was off as the controller was disabled. The signal in blue is the temperature sensor output in Volts, where 0.3V corresponds to 30 centigrade, and the red signal is a linear fit to the data. The x-axes correspond to the time in seconds. For instance this graph shows that when no control action is taken, the fin heats up 0.9 degrees per minute.

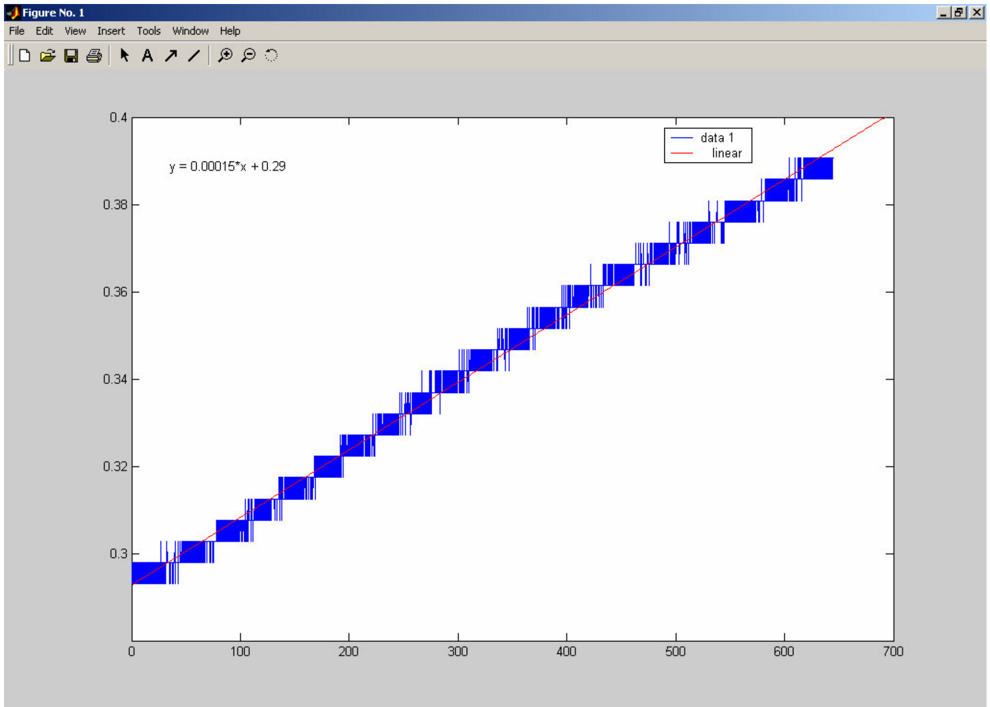


Figure 5.24 Sample temperature sensor log

35. **Clear All:** When this button is activated, the memory of the software is cleared. All variables are reset to their default values. The status bar as well as any other information is also cleared. This indeed, sets the software to its initial form.

36. **Command Area:** This area is intended to enable the students enter Matlab commands as well as execute their own codes from within the GUI. The commands entered in the editor area are executed when the Command button is pressed.

37. **AutoDemo:** The execution of this button causes the software to run automatically, beginning from the temperature sensor data logging to the application of a control strategy.

38. **Info:** An information document is opened when this button is activated. The user can have information regarding to the experiment using the document served.

39. **Close:** The close button closes the experiment software and removes any memory allocation.

In the following part, sample roadmap for executing the experiment is given.

5.2.5 Executing the Heater-Cooler Experiment

The Heater-Cooler experiment may be run in the following order presented.

1. The physical setup is activated. The instructions are provided in the MED CD. This enables the activation of the heating elements and heats up the fin surface.
2. The target is pinged via the Ping Target button. This assures that there is a valid connection between the host and target modules.
3. Using the Build Model button the executable is built and downloaded to the target module.
4. Begin the model execution.
5. Observe the temperature change as the heating elements heat up the fin surface temperature.
6. Enable the control strategies, starting from the On/Off control and then the PI control respectively.
7. The students are encouraged to make controller parameter changes to observe their affects in real-time.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

New technological advancements and current market and product domain conditions require new engineering design approaches to be applied, so that multi-functional, complex and reliable products in reduced time to satisfy market requirements can be designed and developed. This requires mechatronics design and eventually mechatronics education to be applied in real life and become a reality.

This situation has motivated the development of this thesis. In this thesis a Mechatronics Education Desk aiming undergraduate level hands-on mechatronics education is developed. The study under this scope covers; selection of appropriate mechanical, electrical and software prototyping tools and equipment as well as determination of mechatronic building blocks, development of a general purpose low cost interface board, a mobile robot kit and two experimental setups addressing real-time control and mechatronic system interfacing.

The work developed in this thesis is aimed to make a well proposed starting point for mechatronics education and building of a base system (platform) open to extensions and additions in the future. Such a system can therefore maintain itself with ease and can have an automatic development.

The first was to make an exhaustive survey on mechatronics and mechatronic education. The current situation is studied and tools used for this purpose are identified. One main result obtained is that there is no common roadmap proposed or applied in mechatronics education. Generally, actions on determining requirements or problems have been local to a specific mechatronic field. In other words, a systems approach has not been practically applied to mechatronics education. In the thesis, realizing this need, a more general point of view is taken and the developed system, although not complete, handles a wide range of mechatronics education requirements. The scope therefore covers from the selection of basic tools to the development of advanced mechatronic system experiments.

The result of the literature survey shows that facilities and prototyping equipment are indispensable for mechatronics training. Therefore, equipments and prototyping area for the students are determined. Although this equipment has been defined and most of them are in use in different lectures and laboratories, an individual mechatronics design laboratory compromising these parts could not be established in the duration of the thesis. The practical establishment has been left into the future. However, the working area (the WorkDesk), where electrical and software prototyping can be accomplished as well as equipment and tools on this place are well defined.

Another challenge in this thesis has been the selection of software tools, which should be compatible with the hardware also. A general problem, also existing in the industry, is the use of independent and different software tools within departments for the design and development cycle of a product or a system. This indeed results in software and hardware compatibility problems, establishment of a higher number of platforms that create higher costs and is usually prone to errors. To overcome this conflict, software tools available are searched and eventually the application of model based design approach and use of seamlessly integrating software tools is chosen. This has led to the use of model based design software based on Mathworks products, i.e., Matlab and Simulink. After determining this main way, other software and hardware tools are selected to be compatible with each other. The roadmap defined in mechatronics design from this point forward is to make CAD design in Solid Works and then automatically transfer the system (plant) transfer functions into the Matlab/Simulink environment. Another way is to model the mechanical and electrical components using SimMechanics and SimPowerSystem Blocksets, still in the Simulink environment. Thereafter, using Simulink, a non-real time simulation can be run to determine the controller parameters as well as working on before the plant is available in practice. The most advantage is gained when the same model is used in generating code to be run in real-time on a real-time target with the plant in hardware in the loop bases. This way, parameter tuning as well as signal tracing and logging in real-time are made feasible using the same software environment. The selection of such software will likely make seamless integration in the future, facilitate and speed up the development of mechatronics projects using the Mechatronics Education Desk.

Although attention is given to use seamless software tools, the microcontrollers currently in use do not support the Matlab/Simulink and automatic code generation environment. That is models generated in Simulink environment can not be loaded to a microcontroller target used in the education desk. This requires the use of additional software modules corresponding to the processor. However, Mathworks has recently announced new microcontroller target platforms (i.e., Motorola HC12), where models generated in Simulink can be transferred to run on these targets. Therefore, a research can be made on the use of such target platforms and replacement of the current microcontroller boards with the newer ones may be made possible. This replacement will not likely to make any unpredictable results within hardware in the desk as attention has been given to make parts independent from each other.

The selection and interfacing of mechatronic building blocks is another important case. As technology is increasing its acceleration in developing new products, newer and better mechatronic peripherals are coming into the market. Therefore, a continuous search on these blocks is indispensable and students should become familiar with a wider range of components. Although we have tried to make and serve the use of several mechatronic components in the desk, this is not sufficient due the reason mentioned above. Consequently, every project that is likely to be designed and developed with the mechatronics education desk should add newer and different type of components.

The ready2go interface board is designed to support mechatronic building block interfacing and speeding up the process of practicing. The resulting design is enough for the current experiments held under the desk. However, extensions of this board are necessary for further specific applications, such that would require beyond the capabilities of the board, i.e., higher current rating motor drivers. Thus, we recommend the students to design and develop additional modules that can work both together and independently, for specific tasks defined regards to the projects that will be developed. A library of such equipment is essential in developing knowledge base on mechatronic peripheral interfacing.

Robots will call attention of everyone including the students in the future as in today. So, students should be familiar with robot applications such as with the mobile robot kit developed. The mobile robot kit will be a clue tool that will serve in demonstrating integration of mechatronic building blocks.

Two experiments are developed. The modularity of experiments is good practice in showing different and further mechatronic elements. However, Heater/Cooler real-time control experiment lacks the use of a complete model based design approach. A mathematical model corresponding to the plant should be prepared and served to the students to make comparisons of a simulation and real-time hardware in the loop simulation.

The development of the Mechatronics Education Desk is a good starting point for mechatronics education. To further update and improve the features of the desk we recommend the following:

- New experiments covering other physical concepts should be developed. Attention should be paid on integration of software and use of common hardware and software tools.
- Bus communication (i.e., CAN-BUS) and new communication architectures (i.e., wireless communication) should be added as add on modules.
- The growth of the mechatronics education desk should be automated. That is, the flexibility devoted by introducing the current elements should be used as an advantage in returning additional components to the desk. Software and hardware libraries should be formed to let students increase the scope and knowledge gained.
- Web based remote experimentation should be added to enhance experimentation at all times.
- New building blocks should be searched and added to the desk. The interfaces for these should also be developed in parallel.
- Other target platforms such as DSP's compatible with the current model based software should be included in the system.

REFERENCES

[1] Craig, K., Stolfi, F., *Teaching Control System Design Through Mechatronics: Academic and Industrial Perspectives*. Elsevier Science Ltd., 2000.

[2] Siegwart, R., *Grasping the Interdisciplinarity of Mechatronics,* IEEE Robotics & Automation Magazine, June 2001, pp. 27-34.

[3] Johnson, C.W., *Mechatronics Education: Where are we and where are we headed?* Elsevier Science Ltd., 2000.

[4] Akay, A., *The Changing Nature of Design Engineering and Education*, 9th International Machine Design and Production Conference, September 13-15, 2000 METU, Ankara-TURKEY.

[5] Autonomous Systems Laboratory (ASL), Swiss Federal Institute of Technology of Lausanne (EPFL), Switzerland. http://asl.epfl.ch/, August 2001.

[6] The Mechatronics Design Laboratory of Northwestern University, USA. http://mechatronics.mech.nwu.edu/mechatronics/index_content.html, July 2001.

[7] The Mechatronics Laboratory of Massachusetts Institute of Technology (MIT), USA. http://web.mit.edu/2.737/www/, July 2001.

[8] The Undergraduate Teaching Laboratory of the University of Waterloo, Australia. http://sydewww.uwaterloo.ca/UnderGrad/Facilities/facilities_teachinglab.htm, November 2003.

[9] The Mechatronics Laboratory of the University of Utah, USA. http://www.mech.utah.edu/~me3200, November 2003.

[10] Department of Mechanical Engineering, Aeronautical Engineering, and Mechanics, Rensselaer Polytechnic Institute (RPI), USA. http://www.rpi.edu/~craigk/jec1022main.htm, November 2003.

[11] Wild, P.M., Surgenor, B.W., Zak, G., *The Mechatronics Laboratory Experience.* Elsevier Science Ltd., 2000.

[12] Hewit, J., *Mechatronics Design: The Key to Performance Enhancement.* Elsevier Science Ltd., 1996.

[13] Luecke, G.R., *Multi-Tiered Control for Undergraduate Mechatronics.* Elsevier Science Ltd., 2000.

[14] Shetty, D., Kolk, R., Kondo, J., Campana, C., *Mechatronics Technology Demonstrator – An Educational Experience.* Elsevier Science Ltd., 2000.

[15] Kurfess, T.R., Witzel, J.G., *Using Mechatronics in Early Design.* Elsevier Science Ltd., 2000.

[16] Field, S., Meek, S., Devasia, S., *Mechatronics Education in the Department of Mechanical Engineering at the University of UTAH*. Elsevier Science Ltd., 2000.

[17] Kyura N, Oho H., *Mechatronics-an Industrial Perspective*. IEEE/ASME Transactions on Mechatronics, Vol.1, No.1, 1996, pp.10-5.

[18] Ashley, S., *Getting a hold On Mechatronics*. Mechanical Engineering Journal, ASME, May, 1997. http://www.memagazine.org/backissues/may97/features/mechtron/mechtron.html

[19] Rizzoni, G., Keyhani, A., *Design of Mechatronic Systems: An Integrated Inter-Departmental Curriculum*. Mechatronics, Vol. 5, No. 7, July 1995.

[20] Bishop, H.B., Ramasubramanian, M.K., *What is Mechatronics?* The Mechatronics Handbook, CRC Press, 2002.

[21] Isermann, R., *Mechatronic Design Approach.* The Mechatronics Handbook, CRC Press, 2002.

[22] Two-Person Mechatronics Laboratory Station. http://www.rpi.edu/~craigk/jec1022pg1.htm, November 2003.

[23] TCG Solutions Inc. http://www.tcgsolutions.com, July 2001.

[24] Design of Mechatronic System Research Group, Department of Mechanical Engineering, Katholieke Universiteit Leuven (KUL). http://www.mech.kuleuven.ac.be/pma/research/mecha/facilities/default_en.phtml, November 2003.

[25] Feedback Instruments Limited, Park Rd, Crowboroughi E. Sussex, TN62QR, England. http://www.fbk.com, October 2003.

[26] Mechatronic Systems, Inc., P.O. Box 11196, Champaign, IL 61826-1196. http://www.prairienet.org/msi/, November 2003.

[27] dSPACE GmbH, Technologiepark 25, D-33100 Padeborn, Germany. http://www.dspace.de, December 2003.

[28] Quanser 80 Esna Park Drive, Unit #1, Markham, Ontario, Canada L3R 2R6. http://www.quanser.com, December 2003.

[29] Bolton, W., 1998. *Mechatronics - Electronic Control Systems in Mechanical Engineering*, Addison Wesley Longman Limited, England.

[30] Shetty, D., Kolk, R.A., *Mechatronics System Design*, PWS Publishing Company, 1997.

[31] Bradley, D.A., Dawson, D., Burd, N.C., Loader, A.J., *Mechatronics - Electronics in Products and Processes*, 1st Edition, Chapman & Hall, UK, 1991.

[32] Fraser, C., Milne, J., *Integrated Electrical and Electronic Engineering for Mechanical Engineers*, McGraw- Hill Book Company, 1994.

[33] Histand, M.B., Alciatore, D.G., *Introduction to Mechatronics and Measurement Systems*, McGraw- Hill Book Company.

[34] Stadler, W., *Analytical robotics and mechatronics*, McGraw- Hill Book Company, 1995.

[35] Snyder, W.E., *Industrial robots - Computer interfacing and control,* Prentice Hall International, Inc.

[36] Barney, G.C., *Intelligent instrumentation: microprocessor applications in measurement and control*, Prentice Hall.

[37] Auslander, D.M., Kempf, C.J., *Mechatronics, Mechanical System Interfacing*, Prentice Hall, 1996.

[38] Sorli, M., Pastorelli, S., *Hydraulic and Pneumatic Actuation Systems.* The Mechatronics Handbook, CRC Press, 2002.

[39] Fraser, C.J., *Electrical Machines.* The Mechatronics Handbook, CRC Press, 2002.

[40] Chiu, T.C., *Electromechanical Actuators.* The Mechatronics Handbook, CRC Press, 2002.

[41] Bansevicius, R., Tolocka, R.T., *Piezoelectric Actuators.* The Mechatronics Handbook, CRC Press, 2002.

[42] Lyshevski, S.E., *MEMS: Microtransducers Analysis, Design, and Fabrication.* The Mechatronics Handbook, CRC Press, 2002.

[43] Martin, F.G., *Robotic Exploration: A Hands-On Introduction to Engineering.* Prentice-Hall, Inc. Upper Saddle River, New Jersey, 07458, 2001.

[44] Jones, L.J., Flynn, A.M., *Mobile Robots: Inspiration to Implementation.* A.K. Peters, Ltd., 289 Linden Street, Wellesley, MA 02181, 1993.

[45] Pham, H.A., Kua, Y.S., Salem, K., *Having Fun with Muscle Wires.* Elsevier Science Ltd., 2000.

[46] Measurement, Instrumentation and Sensors Handbook, CRCnetBase, 1999.

[47] Lynch, K., Peshkin, M.A., *Linear and Rotational Sensors.* The Mechatronics Handbook, CRC Press, 2002.

[48] Eren, H., *Linear and Rotational Sensors.* The Mechatronics Handbook, CRC Press, 2002.

[49] Norris, P.M., Hosticka, B., *Temperature Measurements.* The Mechatronics Handbook, CRC Press, 2002.

[50] Thorn, R., *Flow Measurements.* The Mechatronics Handbook, CRC Press, 2002.

[51] Jack, H., Sterian, A., *Control with Embedded Computers and Programmable Logic Controllers.* The Mechatronics Handbook, CRC Press, 2002.

[52] The Handy Board, Martin, F.G., http://www.handyboard.com, November 2003.

[53] Grove, Donald, E., *The Vertical Integration of Mechatronics at Virginia Tech,* Thesis, Blacksburg, VA, February 2001.

[54] The Mathworks. http://www.mathworks.com, December 2003.

[55] Mobius Robot, Mechatronics Class. http://www.leang.com/robotics/past/mobius/mobius.html, November 2003.

[56] Boe-Bot, Parallax Inc., http://www.parallax.com, February 2003.

[57] Libby the Robot, Blue Bell Design Inc. Post Office Box 446 Gwynedd Valley, PA 19437-0446 USA. http://www.bluebelldesign.com/Runabot_Detail1.htm, November 2003.

[58] Mech-Bots. http://www.leang.com/robotics/past/mechbots/mechbot.html, November 2003.

[59] iRobot, Corporate Headquarters, iRobot Corporation 63 South Avenue Burlington, MA 01803. http://www.irobot.com, November 2003.

[60] The Palm Pilot Robot Kit. http://www-2.cs.cmu.edu/~reshko/PILOT/, November 2003.

[61] Lynxmotion, Inc. PO Box 818 Pekin, IL 61555-818. http://ww.lynxmotion.com, November 2003.

[62] Crisman, J.D., *Systems Design via Small Mobile Robots*. IEEE Transactions on Education, Vol.39, No. 2, May 1996.

[63] Gamota, G., *Mechatronics in Japan: a Trip Report*, September 1998. http://www.wtec.org/loyola/mechtron/toc.htm, November 2003.

[64] Hewit, J. R., 1994. *Preamble to Mechatronics the basis for new industrial development*, Proceedings of joint Hungarian-British international Mechatronics conference, Budapest. IEEE/ASME Transactions on Mechatronics, Volume 1, Number 1, March 1996.

[65] Mechatronics Research Group, Loughborough University, United Kingdom. http://www.lboro.ac.uk/research/mechatronics/index.htm, November 2003.

[66] Erden, A., Lecture Notes, 2002.

[67] United Kingdom Engineering & Physical Sciences Research Council (EPSRC), http://www.epsrc.ac.uk/website/index.aspx, November 2003.

[68] Day, C. B., "*Mechatronics - meeting the technician's needs into the 21st century*", Conference on Mechatronics - The integration of Engineering Design, Dundee, 1992.

[69] Hewit, J., 1993. *Mechatronics - the contributions of advanced control.* In proceedings of 2nd Conference on Mechatronics and Robotics, Duisburg, Germany.

[70] Rzevski, G., 1995. Mechatronics - Designing Intelligent Machines, Vol. 1, Butterworth-Heinemann Ltd., England.

[71] Auslander, D., *Lecture Notes*, Department of Mechanical Engineering, University of California, Berkeley.

[72] Alptekin, S. E., *Mechatronics Design Studio: Sample Student Projects.* Proceedings of Mechatronics'96, San Francisco, June 13-15, 1996.

[73] College of Engineering, University of Washington, USA. http://mechatronics.ise.polyu.edu.hk/intro.html, November 2003.

[74] Yamaguchi, T., Mechanical Engineering Laboratory, Hitachi Ltd.

[75] Tomizuka M., Department of Mechanical Engineering, University of California, Berkeley, USA.

[76] Electromechanical Systems Laboratory, Department of Mechanical and Industrial Engineering, University of Massachusetts. http://www.ecs.umass.edu/mie/labs/emsl/mechatronics.html, November 2003.

[77] Department of Mechanical Engineering, Mechatronic Engineering and Manufacturing Technology, Chico State University, USA http://www.ecst.csuchico.edu/__depts/mem/BS_Mechatronic_Engineering/index.html, November 2003.

[78] The Robotics and Mechatronics Laboratory, Clemson University, USA, http://ece.clemson.edu/crb/whatis.htm, November 2003.

[79] Pezeshkpour F., *Video for Matlab (VFM),* School of Information Systems, University of East Anglia, Revision 0.1, 1998.

[80] Industrial Research and Development Advisory Committee of the European Community.

[81] Journal of Mechatronics, http://www.elsevier.nl/inca/publications/store/9/3/3/, November 2003.

[82] Mechatronics at Rensselaer, RPI. http://www.rpi.edu/~craigk/, November 2003.

[83] Department of Mechanical Engineering, University of California at Berkeley, USA. http://me.berkeley.edu/mechatronics.html, November 2003.

[84] The Department of Mechanical Engineering, University of Adelaide, Australia. http://www.eng.adelaide.edu.au/mechpages/Courses/What_is_mechatronics/whatis.html, November 2003.

[85] Buur, J.: 1990, *A Theoretical Approach to Mechatronics Design*, PhD Thesis, Institute for Engineering Design, Technical University of Denmark, Lyngby.

[86] Department of Mechanical Engineering, University of Washington, USA. http://www.me.washington.edu/~dig/medesign/Mechatronics.htm, November 2003.

[87] Virginia Polytechnic Institute, USA. http://mechatronics.me.vt.edu/, November 2003.

[88] VxWorks Operating System, WindRiver. www.windriver.com, December 2003.

[89] Telelogic AB. www.telelogic.com, October 2003.

[90] Bloom, B.S., *Taxonomy of Educational Objectives*. New York: David McKay Company Inc., 1956-1964.

APPENDIX A

MED CD

What is Mechatronics?

A literature survey is conducted on the definitions of mechatronics. The results of this survey are provided in the Mechatronics Education Desk Compact Disk (MED CD) under the directory named Mechatronics Education Desk/Appendices/Appendix A. The reference material is also made available in this directory.

Mechatronics Laboratory

The requirement list representing the equipment to be placed in the Mechatronics Design Laboratory housing the Mechatronics Education Desk is given as excel sheets in the MED CD under the directory named Mechatronics Education Desk/MED Laboratory. The equipment are selected to support the education desk equipment and are subject to common use between the members of the laboratory and are to be served as central supplies.

WorkDesk

The type number of the WorkDesk as well as sample views of the desk can be found in the MED CD under the directory named Mechatronics Education Desk/WorkDesk.

RT-TM

The explanations and usage of the software modules developed for use with MED can be found in the MED CD under the directory named Mechatronics Education Desk/RTTM.

The Mobile Robot Kit

Further details of the mobile robot kit, including part lists, assembly instructions, software modules and technical drawings are provided in the MED CD under the directory named Mechatronics Education Desk/Mobile Robot Kit.